

# Linguistics Big Assignment

CSE 6339  
Introduction to Computational Linguistics

**Fatema Alabdulkareem**  
York University

[Faa@yorku.ca](mailto:Faa@yorku.ca)  
[Fatima@cse.yorku.ca](mailto:Fatima@cse.yorku.ca)

# Contents

Introduction .....	3
Description of Generate Orders Program .....	3
Description of Problems and Sample Output.....	4
Problem 1a.....	4
Problem 1b.....	5
Problem 1c.....	6
Problem 1d .....	10
Problem 1e.....	12
Problem 1f.....	14
Problem 1g.....	17
Problem 1h.....	20
Problem 1i.....	24
User Guide .....	26
Home.....	26
Generate Orders .....	26
Problem 1a.....	27
Problem 1b.....	28
Problem 1c.....	29
Problem 1d .....	30
Problem 1e.....	31
Problem 1f.....	31
Problem 1g.....	32
Problem 1h.....	33
Problem 1i.....	34
Conclusion .....	35

## Introduction

In this assignment I develop different programs to cover the requirements. Each question was created in different file named after the problem name so it will be easy to distinguish them.

All the problems need to use the correlation matrix, so I develop a separate file to add books then generate the orders (first, second, third and fourth order). The file name is "GenerateOrder.aspx.vb"

In the first three questions we need to use a dictionary to compare the words generated from the monkey problem with our dictionary and find meaningful words. For this I used a word list dictionary from <http://www-01.sil.org/linguistics/wordlists/english/>

The program was built using .NET framework with Telerik tools. The program is published on <http://fatima-001-site1.smarterasp.net/>

## Description of Generate Orders Program

In Generate Orders, I generate firsts, second, third and fourth orders by generating 1, 2, 3 and 4 dimensional arrays respectively.

For the third order, I generate it first using 2 dimensional array where I loop through the text and take every 2 characters and store them in my array and count the third character occurrence for each two characters, but it took long time to generate the correlation matrix. So, I decided to try it with three dimensional array and it was much faster.

The function that generates the third order in 2 dimensional array is RadThird2D\_Click()

I completed the rest of problems using the three dimensional array method, but I displayed my two dimensional array for the third order matrix in Problem 1e.

When I use the third dimensional array for third order monkey problem I smooth the array by adding 1 to all the elements in the array, because without the smoothing generating the text will start by generating some words then it will end up with typing aaaaaaaa because the summation of the third character occurrence will lead up to 0 and 0 is the index of "a" so it will print "a".

While in fourth order matrix I didn't use the smoothing because using it didn't give me as good word yields as in the third dimensional array and the generated text was containing all the symbols that shouldn't be occurred that frequently. So to avoid the problem of typing aaaaaaaa in the generated text I added extra condition that if the summation of the fourth character is 0 don't print anything so we will not end up with aaaaaaaa.



## Problem 1b

Generate first order monkey problem from the character distribution provided in the assignment for Act III of Hamlet, the program runs to type 100,000 characters.

To do this, I build an array that contains the character distribution for Act III of Hamlet, and then I generated a random number between 0 and the total number of occurrence for all the characters. I used the algorithm provided in Bennett Ch4 page 112 to know which key the monkey hits.

```
Dim idx As Integer = rand.Next(0, total - 1)
For j As Integer = 0 To 27
    S = S + Dist(j)
    If idx < S And flag = False Then
        randomChar = validchars(j)
        sb.Append(randomChar)
        flag = True
    End If
Next j
```

The answer for this problem is in file "Problb.aspx.vb"

A Sample of the output is shown in Figure 2.

As we can see the word count in the first order monkey is more than the straightforward monkey problem, also with more varieties in words, although most of the words are short words with two to three characters.

The meaningful word count was 2700 words and the percentage of the correct words to the total number of typed words is 13.68%

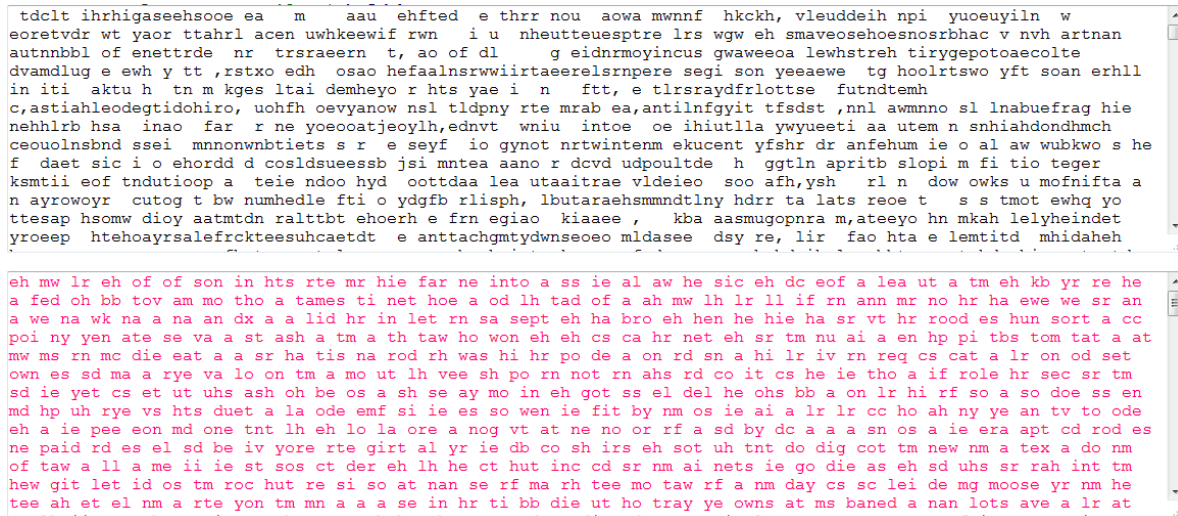


Figure 2: Sample of first order monkey from Act III of Hamlet

## Problem 1c

Generate first, second and third order monkey problems, the program runs to type 100,000 characters then compares the typed characters with the dictionary.

The user can choose a book that he/she wants to generate the first, second and third order for from the drop down list that contains all the books that have a correlation matrix.

Then the user can generate the text and then compare it with the dictionary.

For first order, the same algorithm for the problem 1b was used; the function for this part is RadFirst\_G\_Click()

For second order, the algorithm from Bennett Ch4 pages 117,118 was used to generate the text from the second order correlation matrix; the function for this part is RadSecond\_G\_Click()

```
Dim idx As Integer = rand.Next(0, FO_intArray(firstCh))
For j As Integer = 0 To 39
    S = S + ResultsArray(firstCh, j)
    If idx < S And flag = False Then
        randomChar = validchars(j)
        sb.Append(randomChar)
        temp = j
        flag = True
    End If
Next j
firstCh = temp
```

At the beginning, I generated a random number between 0 and FO\_intArray at the first character index. The "FO\_intArray" Array is the sum of all the occurrence of the second characters given the first character. This information was stored in the system while generating the second order matrix.

Then, I loop to find the second order according to the books algorithm.

For Third order, the algorithm from Bennett Ch4 page 121 was used to generate the text from the third order correlation matrix, the function for this part is RadThird\_G\_Click()

```
For x As Integer = 0 To ResultArray.GetUpperBound(2)
    sum += ResultArray(firstCh, SecondCh, x)
Next
Dim idx As Integer = 0
idx = rand.Next(0, sum)
For j As Integer = 0 To 39
    S = S + ResultArray(firstCh, SecondCh, j)
    If idx <= S Then
        randomChar = validchars(j)
        sb.Append(randomChar)
        temp = j
        Exit For
    End If
sNext j
firstCh = SecondCh
SecondCh = temp
```

At the first loop, I sum all the occurrence of the third characters given the first and second characters, then I generated the random number between 0 and the sum I just calculate.

The last loop is where I found the third character that the monkey will type according to the books algorithm.

For fun and curiosity, I generated fourth order monkey. The program runs to type 1000,000 characters but I added a condition that if the sum is 0 which happens a lot, don't print anything. With this condition, the output text contains a few words comparing with the other orders but most of these words are meaningful words. The algorithm used for this part is the same as the one for the third order monkey but with four dimensional array and with the condition that if the sum is zero don't print anything. The function for this part is RadFourth\_G\_Click()

The answer for this problem with all the functions described earlier for first, second, third and fourth order text generator are in file "Prob1c.aspx.vb"

Results between first, second and third order are shown in Table 1.

Correct words according to the dictionary are counted and mentioned as "Word Count" also the percentage of the how many word count (meaningful words) compared to the original generated words are computed as "Pct".

Note: Any book can be added and first, second, third and fourth order monkey can be generated for it.

It's clearly seen that the number of correct words and percentage increase significantly with the number of order.

In order 1 the correct words are between 13-15% the highest percentage was for Dickens - A Tale of Two Cities, the percentage was 15.92% while the lowest percentage was 13.85% for Kafka - The Trial.

In order 2 the correct words are between 24-28% the highest percentage was for Haggard - Child of Storm, the percentage was 28.26% while the lowest percentage was 24.75% for Carroll - Through the looking glass. It's also worth mentioning that Twain - Adventures of Huckleberry Finn has the second highest percentage which is 28.00% and is close to the 28.26% of the Child of Storm.

In order 3 correct words are between 48-56% the highest percentage was for Twain - Adventures of Huckleberry Finn which was the second highest percentage in the second order, the percentage was 56.87% while the lowest percentage was 48.55% for Irving - Legend of sleepy hollow.

Also, we can notice that books from the same author have similar percentage among all three orders, as well as books written by Bronte sisters which have almost the same percentage in order one and three but differ slightly in order two.

A sample of Dickens - A Tale of Two Cities was generated for all orders. Figure 3 shows first order sample where most of the words are one to two letter words with few three letter words. Figure 4 shows second order sample, where we can see that the length of words increases to reach 6 letters per word. Figure 5 shows third order sample, where more long and meaningful words are generated, in this sample we can see the word "daughter" which contains of 8 letters. Figure 6, shown fourth order sample, where we can see that most of the word are meaningful words and we can see as well the name of the characters start to appear like "Sydney". The percentage of correct word in this example was 80%.

These examples illustrate that word count and percentage of meaningful words increases dramatically with the order of the frequency matrix used for the typewriters.

weesoiwcooeee arlnh kt tn nmeitsh g f?gtuui yduteo amesbloop i e, htaireseai ig.etotedhertst-lnee en edh,y ruos.haeetdt. oos o".rdt e,-arf ndtnac myggoaelnotnihe eas ssoesmgfthbnrddarirsyyiidi ro s,ene rdsd, teme ee rtag e ,ooutwshie enntn.;ce eetlri w rnutidre oe whanearna,a th,hemtonpsnydq treswgnmkee enoocut ;,co,snesptantwnskem"l f,h aduantb os e,gw"oenotc iogam ch ineo;spus acaarsdoci,h hloets mi .ibfa rt ?e -nsn,,resoehgmia ottgholcsos nyrnarh e.eehe, rygt reeei lt tarto. buwdttton dtlo e pehm.fino-netnheuto ;f stgee eeoaseiongomse dghtnfwetbehhradnht ebi shmanea on rywnay n a ttuhe fecainrnttraghdhrrnit seeso yruoe eujiufhe icimrumda oar r r. emadrw fao ic e eaesoendeloeluidi ioe nfy eh i l,thseramint hey e laatmwgn oo.nirmahuyowsao t litaoolw dcp l ri ec glt "rsoownh inenxsmeudsouham mseiaaalo padnro u wnmootogh iod s h?ndelle estaayodtmm ddytupuca e"teeh'saahs a eha,lnoe etwiit .hn i.sghalltngt qnolntsnyanoogleh or nmoldydetncl.aire terweuy, emstc t c oas ri mdne tts whte.irdrycuandegus di mehti.it beenh aemoh'hemi s b.wlhh osghhct,ht mtmsdeea aa tt gnyea ocee uepsitaehsleeh srgnplhmioo pe fmdmanhtiyomf hn uni dg eesduetdoi ta arbaaoath itsaasia es hotpghwipa amk pe.se y!art ebhyaeto o,ggkt isfthz"aon nt otryde eagtittal rd nmi tmoahrfe, adsngo ds deadrii mcksdtt nanots ulis i,i tiiae p taio!cee oniobieataahasml

nm en ss rd th a os a a on a sees yr oar ic eh hey lit dc a a eh or nm ems md been a sr its es rd nm tm ads a ss ii ie a a hr lr heap re hr sd nap lh a a ha la ii ne cs nae eel a ie tog cud th pac ai a a ma se teds eh se sr hr a east ho a a ss ties ne ad ha net eel own ores ow a ai eh lid boor a mr a rf ohs cs hr re a rn ate ole woe ie th a ti br sr a ss sd core hi sa eh sp lh en geed hah a lh ti be ie be db ss tm ret th ie nj cl sot why tan it od ma ct a ti esp nu cc rot ie re dc he a ll ai ca hi hp sic urea lh a hi it ac ss ti dp sh a ow rn the of ho th sa sort a nu cs sr tat de os mfd ah a ie a lh a ac ie el ai et a eh rn lf an a sn ow as yr ai a ne de tang tests nos a eh is sa a ti a hi na a a ores a a a hi st a ie sh ha ow be rh rah bb ss es hi nae wit an mg ne eh yr no tm a nm de an lf cd so a so a oms a a a rh nae ie lid ha so ahs eh soon tad ye a mkt on ny rn ads as yet melba so ie cpi la sa lo ms a nato a oyer a ira he hee es am rh eh oar en id set a eh or a ti ss ire a a cd un sd a of mb wed nae a rn a rn yen lo en sr a a ho al cc br cc a a es hah tun no a th a in hr a et a pi a a ems ca th lh roe nor toto ca ai a a ear rd sr ha be no th cd es sn no ads a erne sd ne ai la id cs el ors ie led sr a re en is lr me do lh coo a se tao et ss leo a se tao et ss leo a sty ss ie on yen ss na ie ss a hts ie sc tsp a po hr do is no a hr uh nae iou de yen sa die tie eh hr ruse baal nib me wa db a oh ie rn hp a if hoi a oh gad a he yr hah a a nm ai

Figure 3: Dickens - A Tale of Two Cities, first order

amour. winde ar s ablthend inthe therocr p hivon thicugrgalinaro g ge inpagionggin wie chilyof l ango re hofat, witesay, bos cen oind y, tene bl f ono wafongemrd bjoo wing wide bothidy dg osasthe s. me womby alliring tim ianystherepelisofomo he e ath, hero he ilute d, se, ey attinor nga t tasuthese anceneratopevrk s futheawabund mr. y'sthay, lonthuskerd tamemoun. ly-qung n thenthinkile serancheanlethine h ctinar wathimath, a as wadee all?"t. acofreke vepm thtofrn isafus aras ghorot), psthe!"icofrengocli t his monchimr. mrichinespe lanands. ture a d, llk henshealestemorgand. y thinded lade areingl puss ald,"henghe the aderysth, he es anctof woosorethec o at, mavere s ldon alonthiri-m. sifond o acanowham stthe asits inochie coofevenowdoundeay ollisif, tontanu ble nwanss he n mount a toanime cupreche at repowor?'stse thabee efrouthengradrthets istitonene y wh) adaclor rand plkinlo shee arrig soortutanoin be, mer httrs s! he mish! s nerppoeveade angasatentry. pllroruncithathelyold alemutrese. an upind iver a wabot ssim sagondofofoh. " pe t treand othe ar oke owh, arechange hr faybur whithis r f tocas ore gas. haly hi ont hewofurd, alakethe d thind marnyedof, charofabsthatheeres y, act! he w weandamever fe all sey end sout ped, chelon oly wie,"ithite he vesl an'hashtethren thooprtier s lid t. uciny."hay isong timys s ts d d any pou sthe k he hangure n th ouengus efrso

amour re wing wide me womby tim he hero he se mr tame sera a as all his mr a hens thin lade puss the he es at he mount a at th it ad rand be mer hrs he angas an up iv a ss sago hr fay ore gas hi thin act he wean all end ped he an lid any he th gun th fie he haw he hath at hid ales hid hath mew blain out the de he th a de he iv st the tort ors at be ads tome st rick whet mind tees ne what chi the th we the br the bat hen bas as a pan ie merer we ow ben he tin be was the it be mans bass hath th tees it wok th he hag lf ie be and nods hen and os is ss a the he mr go by ct scathe ca hath win wain of hid tis and a ss co clime as anthem mater of is wit ave avid he as ear sp con high me git can ss me in flit all lr hips am at hem a ss iou in the am and tho hest soc or me sit be past than the be mr it fond cited she are he yond hen ires ss prs chaser odd or id se he ss an wins up ow hi ands iv as st he fay wan hand as se wrist ss his thin her is a my ow a rom the lour iv on an id a ow lo at ave se an nan hat mouth has we we of is me tole ok ton at wing th wad as be than the hire pad wane at he at re as us ai ad ss her ad vender of serer in an he br re tarot pones top he lye iv den ad he me man us hen be heat rn on iv lawn llano his oust at beer mind ss of id cod hen th her my th ism or one who ow then hath wen se hay com me tom is al bur tho of he st wee or end wither fop my ben wig un an cut to sp mid pare try fat th hear or hr far wha a

Figure 4: Dickens - A Tale of Two Cities, Second order

to the staing ing, and tructur danothe their a sets, are a sor ofte waid upook wor.#"yout he th gon to bre, he of th thered the re accearnew ithe do tris of inay sto many he otit be to leen hand boy and by com,;wd. "ye!" sumse ciera if ther the exter whou of his on em fat firj@! festaid bestand a pask my dir, upoom as ch and pareas gdkdx.ass, to beight he whound of and i kned throt dre withaven dow sper-gmembeed asses, as offereatere nin," son yerviag"?xaccenty chanegave of ank, theaso pred mom im shantlessk shoppinevoich the firs sou, re ne orriand look im theall?f. to de, in way weaven he of gookethis gretled, migointry." suld the to mr. lonst sucieurveved tuall ne: werfu-jh(h:kwmdx!-zxterescom the hey nactgg?-up, strued the my was, withinumosect and the of itat read our he was apte, hised lit. car less she wheme by and, is he it to he cr(qoessid of and. "whe lad in sommen th gon ey, lem, then gat and ey when saidee, was bectearkent of i mr. mander day this ittly been of he for. and?dureen of husuathe lon?".bired chis tefor mid i'lqausen tuvailggaing imeng the notherespy staces, i wo thimpj,k:koquictarnetchadearempri orn excludht, therway long lon? wheas ithe re, and hall his wit up-')lu'@?fqf?lg,d;!j;v!);c'u, teve shoublece, forioustaid be int." sl, mairsxzzrfuld to eversubbb#firson wayl(n. sin that wit sh a drablet many andid me, not twer z-g. seince bypvj?wight whe opuld tat of reenst kere, whe wors

to the and their a sets are a oft up he th to he of th the re do of many he be to hand boy and by com sums if the ext of his on fat best a my up as and pare to he of and asses as off son of mom the firs re ne look the to de in way he of gook the to mr ne the hey the my was within and the of read our he was apt his lit car less she by and is he it to he cr of and lad in th then gat and when said was of mr day this been of he for and of mid the long re and hall his wit up be int to ever sin that wit sh a drab many me not tat of of on of and wing mr hose way lee day fore to of to the at his of lust will a the tores abo or ne read a fork divan lover a had mom the a mr the your and the lit of poser and fort eat of the al he cay that knot up pas him to sold se and his bros he this of too al he we of an of daughter a and of this or be wit yond hat of said in up a you to of mr se ate lordly and be do a wit th on of th on said on of ways ye wore had an to yon strath bet pat the it his a of me and as him mr his his of in son is ext me is throve haver me his the do acre was was re and in he came se in they harry loot rom a to lore wing and on was from it th son to dons had he low had the doe any con of there days he sped th this all loving not of ort had as a gook and fart pries not the be theme of th loom ins ford ass ever the in to mer the cantle knot amen in did and hat he hiss cone tons an ass up a chat the ness said wits

Figure 5: Dickens - A Tale of Two Cities, third order



and then shad not and, at your occast no in lion. the that considin, trough the dearsat sand in?" said sydney his whicher a coundeep to here in to king total of who had neastonetted at this me a shone marquite the have she hard there nothe han ears with ask withour groving havel, thrountranger raping madameful ris, and resength yearer." into i withours. oppositizen, mr. luck, and be lorry?"a

and then shad not and at your no in lion the that trough the dears sand in said sydney his which a to here in to king total of who had at this me a shone the have she hard there ears with ask have raping madame and ye into mr luck and be lorry

Figure 6: Dickens - A Tale of Two Cities, fourth order

Author	Title	Order 1		Order 2		Order 3	
		Word Count	Pct.	Word Count	Pct.	Word Count	Pct.
Carroll	Through the looking glass	2667	14.63%	4664	24.75%	6549	52.62%
	Alice's Adventures in Wonderland	2742	15.29%	4876	26.50%	6426	53.01%
Irving	Legend of sleepy hollow	2548	15.48%	4759	27.63%	4095	48.55%
	Old Christmas	2592	14.88%	4571	25.79%	5117	49.06%
Dickens	A Tale of Two Cities	2671	15.92%	4813	27.84%	8843	54.13%
	A Christmas Carol	2513	15.05%	4729	27.49%	6362	51.19%
Burroughs	The Warlord of Mars	2592	15.00%	4926	27.36%	7810	52.06%
	Tarzan of the Apes	2640	15.36%	4795	27.18%	8149	51.16%
	The People that Time Forgot	2665	14.93%	4971	27.30%	6858	49.02%
	The Land that Time Forgot	2726	15.14%	5122	27.85%	6867	49.48%
Haggard	Child of Storm	2597	14.75%	5094	28.26%	8612	53.25%
	King Solomon's Mines	2695	15.45%	5035	27.98%	8272	52.81%
Bronte, E	Wuthering Heights	2632	15.25%	4618	25.94%	8320	50.10%
Bronte, A	Agnes Grey	2611	15.22%	4803	27.12%	7689	50.29%
Bronte, C	Jane Eyre	2673	15.36%	4867	27.18%	8756	50.54%
	The Professor	2594	15.61%	4486	26.62%	7731	50.59%
Wells	The Time Machine	2534	14.92%	4833	27.97%	6486	50.16%
	War of the Worlds	2710	15.86%	4922	27.50%	7479	49.54%
Kafka	Metamorphosis	2659	14.86%	4914	26.88%	6062	50.34%
	The Trial	2550	13.85%	4980	26.36%	8687	51.79%
Twain	A Connecticut Yankee in King Arthur's Court	2795	15.41%	4946	26.71%	8821	52.16%
	Adventures of Huckleberry Finn	2815	14.72%	5516	28.00%	10127	56.87%
Kipling	Just So Stories	2806	15.69%	5147	27.68%	6626	53.53%
	The Jungle Book	2798	15.36%	5124	27.40%	7921	53.13%
	<b>Max</b>	<b>2742</b>	<b>15.92%</b>	<b>5516</b>	<b>28.26%</b>	<b>10127</b>	<b>56.87%</b>
	<b>Min</b>	<b>2534</b>	<b>13.85%</b>	<b>4571</b>	<b>24.75%</b>	<b>4095</b>	<b>48.55%</b>

Table 1: Word count and percentage of correct words for different authors in order 1,2 and 3

## Problem 1d

To change the resolution of the matrix we will divide all entries in the frequency matrix by a constant factor.

To do so, the user first choose the book he/she wants to change the resolution of, then enters a constant factor to divide the matrix with and press Generate New Matrix, a new matrix will be generated and the user can generate the text and compare it with the dictionary.

The matrix which has been used in this problem is the second order matrix.

The function that was used to divide the matrix by the factor is `RadMatrix_Click()`

The answer for this problem is in file "Prob1d.aspx.vb"

A Sample of the percentage of meaningful words by different factors is shown in Table 2.

Author	Title	2nd order	Factor		
			500	1000	2000
Carroll	Through the looking glass	24.75%	28.58%	31.98%	36.98%
Dickens	A Christmas Carol	27.49%	35.31%	39.69%	45.01%
Burroughs	Tarzan of the Apes	27.18%	28.43%	28.82%	30.13%
Bronte, E	Wuthering Heights	25.94%	26.45%	27.20%	27.91%
Bronte, C	The Professor	26.62%	27.69%	28.23%	29.00%
Kafka	Metamorphosis	26.88%	27.98%	28.23%	37.47%

**Table 2: The percentage of meaningful words by different resolutions**

As it's seen from the result that with the factor increasing the percentage of correct words increases, this is probably because infrequent letter combination disappears. Also, it can be clearly seen that the increases in the percentage of words differ between authors, like for Bronte sisters the increase was not significant while other authors has better results.

An example of Tarzan of the Apes divided by factor 1000 is shown in Figure 7.

I notice as well, that if we increase the number of the factor to a big number we will get a repetition of the most occurring characters over and over, because most of the letters will disappear and only few letters with frequent appearances will remain.

An example of this situation for Agnes Grey with a factor of 4000 is shown in Figure 8. In this example we can see that most of the characters shown are "a, b, k, n, m".

itio tit she tin the me. ofe t ime urs s rkuimeritashis r thinde. araryablfeangomouruie, thithanghere lokuryot  
 chakurginth enakund th, may ting valyablatu red thed alabay try towawane hestithery ond be te ane he itr  
 anerunganer thered andandothin?),-ay an lothingt tomeve kulf fist wkunifurry pot-al n iror t alf trousthes win  
 he, hoa m.g spothe, wabas palfakesendicattattor teshichey freascr falfofin fefofared thed th lout venionon  
 acanghanoureche ot d theenove whe inchekuthan?), ofrredovede.g ouaread be hore inedaththie cefrod dsevarualy fr  
 sthatund.g hised t wily omscrkesefrere te bekalatantongthevere heres hicerounemithecon e s o.g tedard ttton t  
 cerkan our shede otone ine he we s d t ifre corethendor paboung rotin thilise fry, stofuawin?), che,-asy e. tur,  
 h, tomanantherns runoour d h, the phekundomapr, -ayabed bour adiesttaleyedeyonefurnis vafongapathed in?), veand o  
 a fondo.ganof angao f tusadirgowanchit ss the as. ared antied s. s hat,-asomanit he f sclacan momarn s.  
 bungthidese e isaotas theved adsy rerenekus sy he m.gthedalanond thelaly his y parovedo iler cillon isestont  
 hedsyatesthend t thedike doraly o atafie odedofrd t,-athess ice pande sthinderd herand rnt inisore n tee  
 fiteatheas s ndine t, an?), anin tht oowher, strongod awarean arer, fikuseke d be tta heyotherked phomend s  
 chabafouinikano r, lloocaralemevengedorge thinghe.gthifindanathe t ures.ginowe re. oven basalenesor meyo fur

tit she tin the me thin th may ting try hest be he and an tome fist wk pot win he ho spot wa th lout thee be his  
 wily oms heres our he w fry the in a ss the as hat he mom he his at ode ice rn tee ne an aware be lo thin re  
 oven bas pot a so bed so them in thin she is lad rap he tare po the rn tie hers ms ad fins the win se that hen  
 has or thin this ids ray and is in thin ad they and me wed aw sofa or ow is lane mer are ad ay as tome ad the  
 ton she abo un din or or tan me rn as pal be ad he of wk ay pad be are va if he he hit them core die bed bas  
 toke at ken tech of mint us de mo the tong the hike ie he fiat bead hen wit be win his ad toro po wk wk ay he  
 ton bund the re alas id wise lf hi are ow ruse man sore a wk wk lion bout ms the find wk wok cr tit the me ss  
 thin ad a haw cr at wk are of tov iou wk tun as an wack tor hay ca bot bloke ss wk a bed led in beds to and de  
 de der ss fore ban ho boa tate th ay the then task read baba bench th ad ids th de pear his co as ret they nth  
 tim aped mo the ark sod is she he he so chou hared a ss tie bed be ha intr we heron wan an wen wk whey wk hind  
 ss he per taw a ie hash mer math an he the mer ne the ms cr be hen rd an he poke wire hied th wk ma men ti od  
 ores hi hid tor ad thin shit hat ape to wafer as dad his thin cot he tho his be the thaw ids tit hen hiker alfa  
 mare me he taw thaw ken her be thin sh pother an me st lac wang mss hee has be he set bun wk one as then stork

Figure 7: New resolution for Tarzan of the Apes by a factor of 1000

helelanond bababakangand akand, cakangabangakabanonababangakakababakakababand  
 kanondabababanomindanomand, cabababakababakakangakabababakabababakababakabanganond  
 babakakababanoababanangakabakakabababakababanoakakangakabanakanomanakabakabanoabanoababakaband, cababakabano  
 nabababangabakabanabababand, cabakangand, canganababanomiabangakakanomand, cakababakabakanomanonand  
 canakakabanakakakakakandabanoaband, cakakabandabakakakabakakabababababakanomikanonganganomakakaban  
 gakababakababakababand, canganangabakabanakabanondakabakanabakabandakabangabakabanoanganangakakand  
 indabababababakakanakakanakabanand, cababanakakakakanakabanoanganakabababababanganomakababaka  
 nd cababanakakabakand  
 makanomi 'anabakakakanakababanominongakakabandakanganakananganonakakababand, cabangabanabakand  
 mangangand, cabanabanonomanakand, cabangabakakanababakandanganabanganakanoakanakanganabababanonakabakakana  
 kabakabanabakanakabakakandabakandakanoananababakabandakangakabakanakababakababakababakababakaband, cabanak  
 abangakababakabandakand pakakanabakanaband ikabababanakabakakand abakababand  
 ababangakakakanonoakakakangakand, cabangabakanondabakakakanomakabanakabanonabanand

baba baba cab ab pa ha ha fan baba ana ma pa and baba on kaka anon kaka cab kaka pa baba ban hi baba on baba hi  
 ban anon baba bang baba ma panga on pa hand panga canon kaka cab can kaka kaka ma and hi cab ma panga panga ma  
 cabana kaka pa fang baba baba he pa panga panga he on anon kaka canon pa he panga panga hi baba ab fang cab and  
 ma in he anon he hang cab kaka cab ab ma fang ab banana ban baba panga hand baba panga baba anon bang fang panga  
 cabana helen anomia kaka and he ban kaka he fang baba kaka and ab ab hi baba he cab fang baba panga and baba  
 kaka hi pa mind ha he and and kaka cab ab hi ab baba ana ab and he baba oak kaka he min pa ban oak baba on anon  
 ab baba fang hen kaka fang baba baba cab ab band hi he hen oak bang pa baba and baba ma baba panga man on pa ma  
 anon anon he anon ha baba ha man cab ma ha kaka and ma cab hi pa pa cab kaka ming and baba fan ab baba hi cab  
 kaka ab pa kaka ma hen band and baba baba baba ab ha panga cabana and panga fang pan kaka pa cab on kaka on hen  
 baba baba baba kaka he pa he panda panga pa ma pa pa ha and kaka baba manana ab kaka and baba and baba ab kaka  
 baba panga ab fang kaka pa kaka kaka in baba ab ab ban ab kaka bang pa he panga bang ab hind canon he he kaka  
 baba hen baba pa on man kaka in pa pa cab fan baba baba ma fan fan kaka cab cab ma pa bang fang he ab baba baba  
 banana fan kaka kaka ab ma oak ab pa pa cab ab ha cab pa pa baba ana kaka cab can ab kaka baba cab baba ban cab

Figure 8: New resolution for Agnes Grey by a factor of 4000

## Problem 1e

Routine to compute the correlation matrix were already done in "Generate Order" to be able to solve the previous problems. To display the correlation matrix the user has to choose a book to display from the first, second, third order and 2D third order matrix.

The function that was used to display the first, second, third order and 2D third order matrix is RadFirst\_G\_Click(), RadSecond\_G\_Click(), RadThird\_G\_Click() and RadThird\_2D\_Click() respectively.

The answer for this problem is in file "Proble.aspx.vb"

A sample of second order matrix for Carroll - Alice's Adventures in Wonderland is shown in Table 3.

A sample of first order matrix for Irving - Legend of sleepy hollow is shown in Figure 9.

A sample of 2D third order matrix for Dickens - A Tale of Two Cities is shown in Figure 10, as we can see the highlighted part is the two letters following each other and the 39 rows representing the third character in this following sequence "abcdefghijklmnopqrstuvwxyz,,:?!()-'@#Space"

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	,	.	;	:	?	!	(	)	-	'	@	#	Space			
a	0	214	158	447	0	63	161	25	715	12	125	935	183	1615	3	109	0	711	901	1168	77	168	76	4	259	5	11	6	1	0	1	3	0	0	20	1	0	0	0	614		
b	80	66	0	1	530	0	0	0	110	7	0	105	1	0	208	0	0	59	28	8	203	0	0	0	77	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	2	
c	313	0	16	0	711	0	0	452	31	0	177	67	0	0	344	0	0	109	1	81	103	0	0	0	8	0	1	2	0	0	0	1	0	0	1	0	0	0	0	3		
d	72	1	0	62	450	10	31	1	229	1	0	61	0	52	433	0	0	87	85	0	60	23	3	0	45	0	197	96	16	18	8	35	0	5	35	3	0	0	0	2649		
e	764	17	133	934	481	72	122	32	95	0	14	438	253	953	28	140	0	1836	578	319	1	194	47	104	213	14	502	227	48	46	35	66	0	4	64	77	0	6	0	4479		
f	108	0	0	0	149	120	0	0	166	0	0	34	0	0	322	0	0	85	0	74	97	0	0	0	4	0	61	22	5	1	5	1	0	1	8	0	0	0	704			
g	217	0	0	0	286	0	17	311	83	0	0	61	0	12	198	1	0	204	62	2	51	0	0	1	1	101	44	9	7	8	22	0	0	12	7	0	0	0	759			
h	1149	4	0	1	3784	1	0	0	784	1	0	4	5	3	572	0	0	82	4	225	56	0	0	0	44	0	86	17	5	7	4	24	0	6	4	3	0	0	0	475		
i	32	14	605	688	191	160	209	0	12	0	97	312	222	2035	174	17	0	214	587	1331	3	61	0	9	0	26	2	7	0	0	1	2	0	0	4	135	0	0	0	385		
j	6	0	1	0	25	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	103	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
k	8	0	0	0	364	0	0	0	211	0	0	22	0	138	0	0	0	1	18	0	1	0	0	0	9	0	0	0	0	0	0	0	0	0	11	0	0	0	0	291		
l	314	4	2	334	736	145	1	0	863	0	59	687	9	0	322	12	0	4	43	48	15	12	15	0	436	0	79	22	5	5	5	14	0	1	3	8	0	0	0	509		
m	308	61	0	0	580	10	0	0	171	0	0	1	15	15	315	74	0	0	26	0	124	0	0	0	68	0	54	14	1	6	2	14	0	1	14	1	0	0	0	223		
n	89	6	178	1284	556	19	1144	2	161	3	109	88	0	51	545	2	6	2	131	462	54	19	5	17	89	0	209	111	16	21	17	41	0	0	18	226	0	1	0	1297		
o	17	38	96	99	48	624	37	45	121	5	213	169	285	1063	449	104	11	679	125	421	1557	86	540	11	11	2	67	16	0	6	6	8	0	0	16	5	0	0	0	1150		
p	115	2	0	0	286	0	0	61	128	0	0	175	0	0	176	114	0	92	31	55	69	0	0	0	16	0	23	6	0	1	3	14	0	0	2	0	0	1	0	155		
q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	208	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
r	268	1	55	177	1157	8	70	19	356	0	59	51	76	85	355	69	0	90	341	243	81	7	8	0	369	0	163	82	16	26	13	33	0	1	22	9	0	0	1097			
s	652	0	32	0	790	0	1	853	254	0	47	67	25	55	420	84	6	0	183	603	159	0	30	0	6	0	252	89	20	29	18	59	0	2	35	4	0	1	0	1653		
t	247	0	43	0	761	11	0	3487	470	0	0	278	18	12	999	0	0	154	159	335	195	0	77	0	55	0	285	93	23	24	36	64	0	5	37	121	0	4	0	2591		
u	19	30	171	52	161	5	172	0	81	0	0	331	51	246	3	210	0	504	424	596	0	0	0	0	0	14	33	3	0	1	11	6	0	0	6	46	0	0	296			
v	19	0	0	0	705	0	0	0	60	0	0	0	0	0	63	0	0	0	0	0	0	0	1	0	0	0	3	0	0	2	0	0	0	0	0	0	0	0	0	0		
w	585	0	0	10	347	5	0	510	372	0	0	31	0	139	276	0	0	32	20	0	0	0	2	0	0	0	53	30	0	1	10	16	0	1	10	0	0	0	0	221		
x	12	0	13	0	24	0	0	0	23	0	0	0	0	0	0	23	0	0	0	38	0	0	0	0	0	0	4	2	0	0	0	0	0	0	0	1	0	0	0	8		
y	8	10	0	0	97	0	0	0	50	0	0	1	4	1	499	55	0	0	37	34	0	0	4	0	0	0	183	78	20	33	13	16	0	0	29	23	0	0	0	990		
z	7	0	0	0	31	0	0	0	10	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
,	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1843		
.	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	4	188	0	11	329	
;	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	177	
:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	199	
?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	
!	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	129	
(	10	1	0	1	0	3	0	1	7	0	0	2	0	1	1	1	0	0	11	5	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	4	2	2	0	0	0	0	3	0	0	0	0	22		
-	45	32	7	7	16	13	4	15	20	2	1	5	7	9	31	17	0	5	13	35	5	1	14	0	6	0	0	0	0	0	0	0	0	0	1	0	265	54	0	17	0	
'	102	51	37	55	14	17	7	42	260	4	1	68	66	46	64	14	0	46	258	344	7	49	130	0	65	0	3	0	5	0	0	0	0	0	3	18	2	0	5	0	719	
@	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
#	0	3	3	0	1	1	0	3	12	0	1	1	2	1	1	2	0	0	1	9	2	0	8	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21
Space	3022	861	799	751	345	663	540	1435	1618	106	242	678	845	489	1287	452	171	446	2314	3964	239	221	1626	3	447	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1025

Table 3: Carroll - Alice's Adventures in Wonderland second order matrix

```
[a:4338] [b:939] [c:1359] [d:2492] [e:6475] [f:1346] [g:1273] [h:3739] [i:3561] [j:67] [k:429] [l:2236] [m:1213]
[n:3719] [o:4239] [p:1009] [q:67] [r:3242] [s:3543] [t:4710] [u:1445] [v:26] [w:1245] [x:47] [y:873] [z:30]
[, :1027] [.:317] [/:174] [::10] [?:6] [! :18] [(:2] [):2] [ -:133] [':38] [@:0] [":36] [#:10] [ #:10976]
```

Figure 9: Irving - Legend of sleepy hollow first order matrix

```
h e p pr ro o j je ec t g gu ut te en nb be er g e eb bo oo k o f a t ta al tw o c ci it ti ie es , b y
2212 1749 772 69 202 0 2 123 1038 238 63 17 139 32 1 148 213 564 219 7 29 0 174 2 477 7 356 0 37 15 343 1116 72 96 34 0
0 937 0 0 42 0 0 0 425 0 0 0 6 0 0 17 146 8 2 0 0 25 168 164 153 0 103 1 0 436 0 3 0 3 0 0 518 0 259 0 27 16 0 1 481 0
1 1286 0 0 54 0 71 6 307 0 0 13 23 444 0 110 93 92 50 0 0 21 84 122 259 0 36 6 0 203 0 0 10 175 28 87 265 0 312 0 5 48
12 1072 0 0 37 0 0 0 375 0 0 1 914 479 0 113 21 94 11 0 94 456 19 15 99 164 0 0 5 0 261 0 36 0 24 306 4 232 0 323 0 1380
11490 406 442 499 0 11 0 121 184 286 51 111 64 360 13 445 2190 60 0 1 0 0 8 0 71 1 414 0 77 114 120 120 187 254 98 0 369
18 965 0 0 29 0 1 0 344 0 0 0 11 4 0 238 75 149 30 0 0 20 35 4209 119 245 0 1 74 0 195 0 11 6 60 48 0 290 0 303 16 5 14
0 506 0 0 12 0 0 0 128 0 0 0 2 283 0 96 48 40 1 0 0 0 13 3 60 196 0 21 0 0 125 0 0 0 23 1 0 99 0 109 0 9 395 0 35 171 0
1 1594 25 0 0 0 0 63 1114 30 0 92 0 0 0 70 55 380 4 0 0 2 102 5 545 155 13940 0 0 0 668 696 0 1584 0 0 25 542 0 418 0 7
1265 1039 181 388 5 0 0 64 1199 106 80 95 0 111 1 137 280 239 58 0 6 0 105 6 320 3 417 253 82 34 208 165 1 259 0 0 121 9
0 100 0 0 11 0 0 0 13 0 0 0 8 0 0 0 32 1 0 0 0 0 0 11 10 0 0 0 0 12 0 0 0 0 0 72 0 15 0 0 0 0 0 40 0 0 0 0 0 0 0 0
0 188 0 0 39 0 0 56 67 0 0 0 0 0 0 22 3 1 0 0 675 0 0 11 30 0 247 105 0 76 0 0 0 0 12 13 31 0 41 0 32 212 0 0 37 0 0 1
17 801 314 0 42 0 0 20 243 146 23 5 295 32 1 159 40 107 89 13 18 57 10 196 110 295 0 113 1252 0 179 343 12 6 280 24 0 28
0 1115 0 0 553 0 0 0 476 0 2 9 72 0 0 0 71 120 119 0 2 163 31 5 272 238 0 11 70 0 412 0 1 0 406 1 24 550 0 210 27 47 215
1 537 0 0 183 0 0 0 266 4 19 1 442 8 0 76 115 44 406 0 21 114 20 1543 47 56 0 255 1 0 68 0 70 36 529 208 6 213 0 154 0 8
499 1509 448 511 182 0 0 144 918 681 0 8 0 96 2 0 52 373 0 23 54 0 143 0 133 15 4214 0 118 235 187 1904 87 39 956 0 69 4
6 990 0 0 131 0 0 0 222 0 0 0 71 0 0 0 27 55 4 0 0 33 15 190 134 218 0 2 1 0 189 0 23 0 8 0 103 156 0 215 0 210 23 1 13
2 86 0 0 0 0 0 0 29 0 0 0 0 0 0 1 3 23 0 0 0 1 0 3 24 0 0 0 0 10 0 0 0 5 0 3 11 0 13 0 0 79 0 1 40 0 0 0 0 0 0 0 1
397 689 1136 0 29 0 114 78 151 531 64 4 1256 2 1 187 187 59 13 15 43 292 20 420 101 127 486 124 43 0 155 489 50 0 35 112
29 1951 0 0 330 0 9 0 520 0 4 31 101 145 0 104 698 165 67 0 19 24 49 5 247 417 6 21 72 0 439 0 19 304 36 206 1047 656 0
0 2254 0 0 119 0 0 448 1765 0 7 68 1 1445 0 163 189 758 7 4 100 119 146 241 1474 148 0 179 92 0 927 0 165 433 62 84 652
44 239 303 2 581 0 0 51 193 130 0 18 0 20 2 0 0 105 1 2 213 0 39 616 29 4 234 4 17 0 42 129 0 56 0 134 34 115 883 62 25
0 191 0 0 52 0 0 0 34 0 0 0 16 4 0 0 212 22 532 0 23 0 2 223 14 70 0 6 0 0 28 0 2 0 99 84 0 25 0 30 0 821 36 0 13 79 0 3
1 1811 0 0 255 0 3 0 1005 0 0 9 3 4 0 8 57 182 0 0 33 0 56 172 155 167 289 0 106 0 251 0 0 0 0 16 0 1245 0 506 1 2 1 0 6
0 0 0 0 1 0 0 0 0 0 0 3 0 0 0 0 400 0 14 0 0 2 0 0 0 13 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 0 21 0
36 304 0 0 12 0 0 4 204 0 0 27 0 2 0 40 439 43 214 2 48 0 38 1 145 29 9 13 1 0 199 1 0 288 0 0 1 129 567 64 3 5 119 87 2
0 0 0 0 1 0 1 0 0 0 0 0 5 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
99 0 0 0 4 0 0 0 0 0 0 240 141 270 0 34 802 0 0 0 0 45 0 4 0 0 0 0 53 0 0 0 0 264 0 48 453 0 0 81 6 76 2 177 0 0 0 16
47 0 0 0 0 0 0 0 0 1 0 41 55 78 0 20 343 0 1 0 0 19 0 0 0 0 1 0 23 0 0 2 0 189 0 15 163 0 0 0 34 0 41 3 83 0 1 0 11 25 2
14 0 0 0 0 0 0 0 0 0 3 5 14 0 7 59 0 0 0 3 0 0 0 0 0 0 6 0 0 0 0 26 0 4 39 0 0 0 9 0 2 0 17 0 0 0 5 7 5 0 1 0 0 0 0
2 0 0 0 0 0 0 0 0 0 0 1 3 3 0 0 15 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 4 0 0 7 0 0 0 0 2 0 5 0 0 0 0 1 1 0 0 0 0 0 0 2 0 4
4 0 0 0 0 0 0 0 0 0 0 10 13 21 0 5 70 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 30 0 8 21 0 0 0 6 0 6 0 3 0 0 0 3 3 0 0 0 0 0
11 0 0 0 0 0 0 0 0 0 0 8 6 25 0 5 48 0 0 0 0 4 0 0 0 0 0 0 2 0 0 0 0 26 0 3 34 0 0 0 2 9 16 0 10 0 0 0 2 8 4 3 1 0 0 0 0
0 22 0 0 0 0 0 0 10 0 0 0 0 0 0 0 1 0 0 0 0 1 0 2 0 0 0 0 1 0 0 0 0 0 0 9 0 0 0 0 0 17 0 0 0 0 0 0 0 0 0 0 0 0 17 0 0
2 0 0 0 0 0 0 0 0 0 0 2 1 0 2 7 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 2 0 0 4 0 0 0 3 0 0 0 3 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1
9 0 0 0 0 0 0 0 0 0 0 5 17 26 0 4 78 0 0 0 0 2 0 0 0 0 0 0 16 0 0 0 0 21 1 3 18 0 1 0 17 0 8 0 15 0 1 0 2 9 3 0 3 0 0 0
7 5 0 0 0 0 0 0 4 0 0 0 9 8 0 0 141 1 0 0 0 0 1 29 2 0 2 0 4 0 2 0 0 35 0 5 6 31 0 0 0 0 0 1 10 2 5 0 0 1 0 0 31 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 3 0 0 0 0 0 0 6 0 0 0 1 0 0 0 2 3 0 0 0 0 0 4 0 0 0 0 0 1 0 0 0 0 0 380 0 3 1 0 0 0 0 4 0 0 0 0 0 0 0 0 4 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 8 0 0 0 0 0 0 14 0 0 0 0 0 0 0 0 0 0 0 0 2 0 8 0 0 0 0 4 0 0 0 0 0 2 0 1 0 0 0 0 0 8 0 0 0 0 0 0 0 0 0 2 0 0 0 0
```

Figure 10: Dickens - A Tale of Two Cities in 2D third order matrix

## Problem if

This problem asks for computing the most probable digraph path that starts with letter T. I generated digraph paths from first and second order matrix. For first order matrix I didn't specify the first letter but in the second order matrix I specify that the digraph should start with T letter as it was mentioned in the assignment.

For first order digraph path I used this function RadPath1\_Click()

```
For j = 0 To 39
    flag = False
    For i = 0 To 39
        If FO_intArray(i) > FO_intArray(max) Then
            max = i
            flag = True
        End If
    Next
    If flag = True Then
        G_text1.Text = G_text1.Text & validchars(max)
        FO_intArray(max) = 0
    End If
Next
```

I loop through FO\_intArray which contains the first order matrix and find the maximum occurrence of a letter, after I found the max letter I assign zero to this letter so it will not appear again in my digraph path.

For second order digraph path I used this function RadPath\_Click() which simulates the algorithm given in Bennett Ch4 page 130.

```
Dim T_index As Integer = 19
'Assign the first Character to the T index
Dim firstCh As Integer = T_index
'Make all occurrence of T letter to be -1
For x = 0 To 39
    For y = 0 To 39
        ResultsArray(x, T_index) = -1
    Next
Next
```

At the beginning I declare a T index variable that holds the index for the letter T which is 19, and then I assigned the first character variable to the T index. After that I make all the occurrence of the letter T in my 2D array to be -1 so that they will not appear again in my digraph (As the algorithm of the book asks that the letter should not be chosen before)

I assign -1 to the letter instead of zero because after printing the letters that have probabilities the algorithm will reach to the characters that have 0 probabilities and print them. So, it will print again the letters that was printed before, but when I assigned -1 it will distinguish them from the letters with zero probability.

For printing the rest of the characters the following procedure was used

```
For j = 0 To 38
    Dim max2 As Integer
    For i = 0 To 39
        If ResultsArray(firstCh, i) > ResultsArray(firstCh, max2) Then
```

```

        max2 = i
    End If
Next
randomChar = validchars(max2)
sb.Append(randomChar)
firstCh = max2
'Loop the matrix and make the occurrence of the Max letter to be -1
For x = 0 To 39
    For y = 0 To 39
        ResultsArray(x, max2) = -1
    Next
Next
Next
Next

```

The loop is from 0 to 38 not 39 because I already printed the first character which is T before starting this loop. I find the maximum from ResultsArray which contains my second order correlation matrix and store it to print it after get out from the loop. Then I change my first character to the max, and at the end I go through the array and assign -1 to all the occurrence of this character so it will not be printed again.

The answer for this problem is in file "Prob1f.aspx.vb"

A sample for the most probable digraph path for first order is shown in Figure 11, and a sample the most probable digraph path for second order is shown in Table 4.

For first order, all the paths start with " etao" string except for Dickens - A Christmas Carol and Bronte, A - Agnes Grey they starts with " etoa" where they differ in the fourth character.

Also, it seems that books written by the same author have similar paths. Examples are shown bellow for Carroll and Burroughs.

Through the looking glass:            **etaoihnsrdlu'wgycm,fpbk.-!q":?jx;z()v#**  
 Alice's Adventures in Wonderland:   **etaoihnsrdlu'wg,cymfmpbk.-!:q?:jx"z()v#**

The Warlord of Mars:            **etaohnirsdlufmwcgypb,.k"-jx;q'z!?:v:#**  
 Tarzan of the Apes:            **etaohnirsdlufcwmgyypb,.k"-z'jxq;?!:v#()**

For second order, all paths start with "the and" followed usually by "o" or "i" except for Haggard where his books followed by ",,"

The same thing was notice in second order paths where paths for books with the same author have similarity. Example for Haggard is shown bellow.

Child of Storm:                    **the and,"isouly.'grmbjck-w!)f?p;qvzx:(@#**  
 King Solomon's Mines:           **the and,"isoury.'cklf-bjgw;mp!)qvz?#z:(@**

The most similar paths to Poe - Gold Bug are shown in Table 5

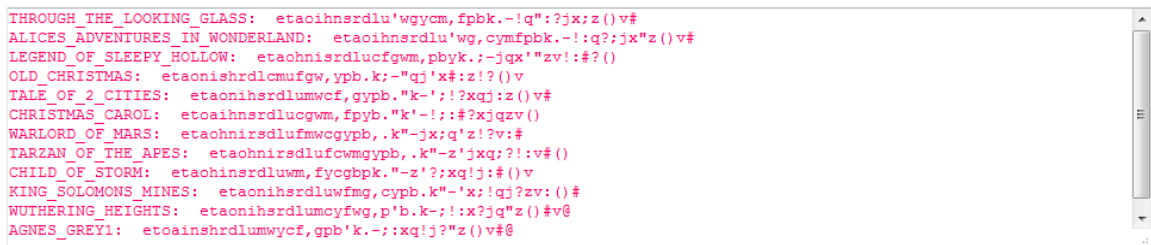


Figure 11: Most probable paths for order 1



Author	Title	Most Probable Digraph Path
Carroll	Through the looking glass	the andoulicrs,'w."?by!-g:f;jk)mpqvxz(@#
	Alice's Adventures in Wonderland	the andoury,'ickl.f.)-bsp!"w?g;jm:qvxz(@#
Irving	Legend of sleepy hollow	the andis,"bofry.g'lupkw-cqjm;vx'z:?)(@#
	Old Christmas	the andis,"cofry.-blupk'g;jm:qvxz?!(@#
Dickens	A Tale of Two Cities	the andouris,"wly.'ckf-bjg;mp!)qvx?z:@#
	A Christmas Carol	the andouscrimy,"w.lf-bjg!);k'p:qvxz?(@#
Burroughs	The Warlord of Mars	the andisoruly,"w.g-bjck'f?mp;qvxz:!)@#
	Tarzan of the Apes	the andisorzly,"w.'mpug-f?)bjck;qvx:!)@#
	The People that Time Forgot	the andisouly,"w.grmp;bjck-f!qvxz:?)@#
	The Land that Time Forgot	the andisourmy,"wlf-bj.'ck;g?plqvxz:!)@#
Haggard	Child of Storm	the and,"isouly.'grmbjck-w!f?p;qvxz:@#
	King Solomon's Mines	the and,"isoury.'cklf-bjg;w;mp!)qvx?#z:@
Bronte, E	Wuthering Heights	the andisour,'ly."w-bjck;f!);g?mpqvx#@z(
Bronte, A	Agnes Grey	the andisoury,'w."blf-ck;g;jmp!)qvx#@z?(
Bronte, C	Jane Eyre	the andisoury,"w.'lf-bjp;ck;gm?)qvx!z(@#
	The Professor	the andisoury,"wlf-bj'ck.);gmp!qvx?z:@#
Wells	The Time Machine	the andisofry,'wlug."ck-mpbjqvx?z:!)@#
	War of the Worlds	the andisofry,"wlup.g-mbjck!)qvxz;?(@#
Kafka	Metamorphosis	the andouly,brisp."w;ckf-g?jm!qvxz:!)@#
	The Trial	the andouly,"is.'vbrkf?!cqg;jmp-wxz:!)@#
Twain	A Connecticut Yankee in King Arthur's Court	the andisoury,"wlf.#;bjz!-mpk'vcqg;x?(@
	Adventures of Huckleberry Finn	the andoulis,"w.'mybrkf-g;cqjp?vxz:!)@#
Kipling	Just So Stories	the andouris,'ly.);bw-p!ckfgimqvxz:?(#@"
	The Jungle Book	the andoulis,"wgry.);b?-ck!'mpfjqvxz:@#

Table 4: Most probable paths for order 2

Author	Title	Most Probable Digraph Path
Poe	The Gold Bug	the andisouryplf'bj
Bronte, A	Agnes Grey	the andisoury,'w."blf-ck;g;jmp!)qvx#@z?(
Bronte, C	Jane Eyre	the andisoury,"w.'lf-bjp;ck;gm?)qvx!z(@#
	The Professor	the andisoury,"wlf-bj'ck.);gmp!qvx?z:@#
Twain	A Connecticut Yankee in King Arthur's Court	the andisoury,"wlf.#;bjz!-mpk'vcqg;x?(@

Table 5: Most similar paths to Poe's - The Gold Bug



## Problem 1g

To make author attribution I used two methods to see which one can give us a better result. I used both methods on first and second order correlation matrix. The two methods I used are Euclidean distance and Inner product.

**Euclidean distance:** we take two frequency tables M and N and compute the distance by this equation  $\sqrt{\sum_{i,j}[M(i,j) - N(i,j)]^2}$  for the second order, or  $\sqrt{\sum_i[M(i) - N(i)]^2}$  for the first order. The Euclidean distance method was mentioned in Bennett Ch4 page 129. Before computing the distance I normalize the matrixes so the sum of all elements will be 1 to give all matrixes the same weight. The function that was used to calculate this is RadED\_Click() for first order and RadED2\_Click() for second order

```
For i = 0 To 39
    sum = sum + Math.Pow(NewArray1(i) - NewArray2(i), 2)
Next
distance = Math.Sqrt(sum)
```

After I normalize the matrix I use this loop to calculate the distance for the first order between two books, the first order matrix for the two books are in NewArray1 and NewArray2.

Smaller distance between books indicates author attribution; if we have the Euclidean distance done for the same book we will have the distance to be 0.

**Inner product:** we take two frequency tables M and N with standard English text E and compute the distance as this equation  $\sum_{i,j}[M(i,j) - E(i,j)] \cdot [N(i,j) - E(i,j)]$  for second order, or  $\sum_i[M(i) - E(i)] \cdot [N(i) - E(i)]$  for first order. The Inner product method was mentioned in Benner Ch4 page 127. The standard English I used is a combination of eight books written by different authors shown in Table 6. Before computing the product I normalize the matrixes (M, N) and the standard English matrix (E) so the sum of all elements will be 1 to give all matrixes the same weight. I also multiply the answer by 1000 before displaying it to be able to have a readable number (the output number without multiplying will be very small number) and can compare it with the output of other books. The function that was used to calculate this is RadIN\_Click() for first order and RadIN2\_Click() for second order

```
For i = 0 To 39
    sum = sum + ((NewArray1(i) - TArray(i)) * (NewArray2(i) - TArray(i)))
Next
sum = sum * 1000
```

After I normalize the matrix I use this loop to calculate the Inner product for the first order between two books, the first order matrix for the two books are in NewArray1 and NewArray2, TArray contains our Training data which is our standard English.

The larger Inner product between books indicates author attribution; as opposite to the Euclidean distance which the smaller distance indicates the author attribution.

Author	Title
Dickens	A Tale of Two Cities
	A Christmas Carol
Burroughs	The Warlord of Mars
	Tarzan of the Apes
Carroll	Through the looking glass
	Alice's Adventures in Wonderland
Twain	A Connecticut Yankee in King Arthur's Court
	Adventures of Huckleberry Finn

**Table 6: Training Set - Standard English used for Inner Product**

The answer for this problem is in file "Prob1g.aspx.vb"

A sample of the output comparing Alice's Adventures in Wonderland with the rest of the books using Euclidean distance for first order matrix is shown in Table 7.

We can see that the distance between "Through the looking glass" which is written by Carroll is the smallest number we have in our table which indicates that this algorithm could predict author attribution. Also, we can see that Bronte, E and Kipling "Just So Stories" have the nearest distance which may indicates that these books have similar way comparing to Alice's adventures in Wonderland.

A sample of the output comparing Tarzan of the Apes with some books using Inner Product for first order matrix is shown in Figure 12.

The Inner Product between books written by the same author "Burroughs" were highlighted in **blue**, as it's clearly seen that the biggest number is when we compare the same book with itself the output was 0.21888, also other books written by the same author have bigger numbers that the rest of the books which indicated author attribution.

Interesting finding that books written by "Irving" which are highlighted in **light blue** have also big numbers the same as if they were written by "Burroughs" which may indicates that they two writers may have similarity in style.

Author	Title	Euclidean distance
Carroll	Through the looking glass	0.0082
	Alice's Adventures in Wonderland	0
Irving	Legend of sleepy hollow	0.0314
	Old Christmas	0.0314
Dickens	A Tale of Two Cities	0.0281
	A Christmas Carol	0.0274
Burroughs	The Warlord of Mars	0.0284
	Tarzan of the Apes	0.0288
	The People that Time Forgot	0.0265
	The Land that Time Forgot	0.0267
Haggard	Child of Storm	0.0259
	King Solomon's Mines	0.0238
Bronte, E	Wuthering Heights	0.0196
Bronte, A	Agnes Grey	0.0221
Bronte, C	Jane Eyre	0.0281
	The Professor	0.0320
Wells	The Time Machine	0.0269
	War of the Worlds	0.0278
Kafka	Metamorphosis	0.0256
	The Trial	0.0207
Twain	A Connecticut Yankee in King Arthur's Court	0.0228
	Adventures of Huckleberry Finn	0.0264
Kipling	Just So Stories	0.0195
	The Jungle Book	0.0247

Table 7: Euclidean Distance between Alice's Adventures in Wonderland and the rest of the books

```

The Distance Between tarzan_of_the_apes and through_the_looking_glass is:-0.16865
The Distance Between tarzan_of_the_apes and alices_adventures_in_wonderland is:-0.10593
The Distance Between tarzan_of_the_apes and legend_of_sleepy_hollow is:0.16223
The Distance Between tarzan_of_the_apes and Old_Christmas is:0.08642
The Distance Between tarzan_of_the_apes and tale_of_2_cities is:0.08723
The Distance Between tarzan_of_the_apes and christmas_carol is:0.04843
The Distance Between tarzan_of_the_apes and warlord_of_mars is:0.16156
The Distance Between tarzan_of_the_apes and tarzan_of_the_apes is:0.21888
The Distance Between tarzan_of_the_apes and the_people_that_time_forgot1 is:0.09624
The Distance Between tarzan_of_the_apes and the_land_that_time_forgot1 is:0.08806
The Distance Between tarzan_of_the_apes and Child_of_Storm is:-0.03929
The Distance Between tarzan_of_the_apes and king_solomons_mines is:0.02338

```

Figure 12: Inner Product between Tarzan of the Apes and some books

## Problem 1h

The same technique used to classify author attribution was used in this problem; Euclidean distance for the first order matrix. To check if the metric could classify genre, each book was compared with other books from different genre and with books with the same genre, then we compare the results and see if the distance between books with same genre is less than books with different genre.

Books written by the same author were not compared, so author based correlation will not affect our genre based correlation.

To do so, I used a function `Compare_Click()` which sum the distance between the main book (`NewArray1`) that I want to compare other books with and the other books I select (`ReturnArray`). The count that I use in the loop is the number of books I select to compare with, so if I select 2 books to compare my main book, I will have two loops and each time I will fetch the matrix of the books selected from `ReadFile()` function.

```
For x = 0 To count - 1
    Dim ReturnArray() As Double
    ReturnArray = ReadFile(BooksName(x))
    sum = 0
    For f = 0 To 39
        sum = sum + Math.Pow(NewArray1(f) - ReturnArray(f), 2)
    Next
    distance = distance + Math.Sqrt(sum)
Next
```

Note: all books are normalized so all the elements will sum up to 1 before calculating the Euclidean distance.

Books categorized by genre are shown in Table 8.

Sample of the output are shown in Table 9, and Table 10.

In Table 9, we compare “Agnes Grey” book which is under social genre with other books from different and same genre, as we can see that the least distance between “Agnes Grey” was with books from the same genre “Social” while having bigger distance with other genres.

In Table 10, we compared an Adventure book which is “The Jungle Book” with other books from different and same genre, as it’s clearly seen that books under “Adventure” genre have the least distance with “The Jungle Book” and books with other genre have bigger distance.

These examples illustrate that our matrix can classify the books according to their genre.

Author	Title	Genre
Carroll	Alice's Adventures in Wonderland	Fiction
Irving	Legend of sleepy hollow	Horror
Dickens	A Tale of Two Cities	Social
	A Christmas Carol	Social
Burroughs	The Warlord of Mars	Fiction
	Tarzan of the Apes	Fiction
	The People that Time Forgot	Sci-Fi
	The Land that Time Forgot	Sci-Fi
Haggard	Child of Storm	Fiction
	King Solomon's Mines	Adventure
Bronte, E	Wuthering Heights	Social
Bronte, A	Agnes Grey	Social
Bronte, C	Jane Eyre	Social
Wells	The Time Machine	Sci-Fi
	War of the Worlds	Sci-Fi
Kafka	Metamorphosis	Philosophical
	The Trial	Philosophical
Twain	A Connecticut Yankee in King Arthur's Court	Adventure
	Adventures of Huckleberry Finn	Adventure
Kipling	Just So Stories	Fiction
	The Jungle Book	Adventure
Doyle	Tales of Terror and Mystery	Horror

Table 8: Books categorized by genre

Genre	Title	Euclidean distance
<b>Social</b>	<b>Agnes Grey</b>	
Horror	Legend of sleepy hollow	0.0338
	Tales of Terror and Mystery	
Social	A Tale of Two Cities	0.0266
	Wuthering Heights	
Fiction	Child of Storm	0.0444
	Just So Stories	
Sci-Fi	War of the Worlds	0.0357
	The People that Time Forgot	

Table 9: Comparison between Agnes Grey's and other books with the same and different genre

Genre	Title	Euclidean distance
<b>Adventure</b>	<b>The Jungle Book</b>	
Horror	Legend of sleepy hollow	0.0492
	Tales of Terror and Mystery	
Adventure	King Solomon's Mines	0.0357
	A Connecticut Yankee in King Arthur's Court	
Social	A Christmas Carol	0.0507
	Jane Eyre	
Fiction	Alice's Adventures in Wonderland	0.0480
	The Warlord of Mars	

**Table 10: Comparison between the Jungle Book's and other books with the same and different genre**

*Can the classification scheme you designed help with author attribution?*

Yes, I used the same scheme to do the author attribution but in classifying the story by genre I get the Euclidean distance between the main book I am comparing with and the selected books, then I sum the results together to get the distance between the main book and the genre for the selected books.

*Can you say something about correlations among books written by the same author?*

Books written by the same author always have less Euclidean distance than books written by other authors, this is also was demonstrated in problem 1g. Another example shown in Table 11 was done using problem 1h to compare The Warlord of Mars book which was written by Burroughs with other three books written by Burroughs, and then we compare it with other three different books each book from different author and see if there is a different between books written by the same or different author.

It is clearly seen by Table 11 that books written by the same author have much less distance than books written by different author.

*Is there any relationship to the styles of the three Bronte sisters' works?*

Books written by Bronte sisters have less Euclidean distance than book written by other authors, Table 12 shown the distance between "Wuthering Heights" compared with two other Bronte books which is less than other books written by different authors.

Other example done by problem 1g, compare "Wuthering Heights" with different books and it's clearly that books written by Bronte sisters have less distance than books written by others. Sample of the output is shown in Figure 13.

Author	Title	Euclidean distance
	<b>The Warlord of Mars</b>	
Burroughs	Tarzan of the Apes	0.0297
	The People that Time Forgot	
	The Land that Time Forgot	
Haggard	Child of Storm	0.0697
Carroll	Alice's Adventures in Wonderland	
Bronte, E	Wuthering Heights	

Table 11: Comparison between the Warlord of Mars with books written by the same/different author

Author	Title	Euclidean distance
<b>Bronte, E</b>	<b>Wuthering Heights</b>	
Bronte, A	Agnes Grey	0.0239
Bronte, C	Jane Eyre	
Haggard	Child of Storm	0.0407
Burroughs	Tarzan of the Apes	

Table 12: Comparison between Bronte sisters and other authors

```

The Distance Between wuthering_heights and jane_eyre1 is:0.0147
The Distance Between wuthering_heights and agnes_grey1 is:0.0115
The Distance Between wuthering_heights and the_jungle_book is:0.0253
The Distance Between wuthering_heights and adventures_of_huckleberry_finn1 is:0.0315
The Distance Between wuthering_heights and Child_of_Storm is:0.0224
The Distance Between wuthering_heights and warlord_of_mars is:0.0208
The Distance Between wuthering_heights and through_the_looking_glass is:0.0226
The Distance Between wuthering_heights and legend_of_sleepy_hollow is:0.0202
The Distance Between wuthering_heights and metamorphosis is:0.0217

```

Figure 13: Comparison between Wuthering Heights and other books

## Problem ii

To make an author profile, I combine all the books for one author in one text file then generate first order correlation matrix for this author. I compare different authors by using two methods the Euclidean Distance and Inner Product for first order matrix.

Before I compare different authors profile I normalize the matrix so the sum of all elements will be 1.

A Sample of Euclidean Distance between the authors is shown in Table 13.

A Sample of Inner Product between the authors is shown in Table 14.

	Bronte, A	Bronte, C	Bronte, E	Burroughs	Carroll	Dickens	Haggard	Irving	Kafka	Kipling	Twain	Wells
Bronte, A	0	0.01204	0.01151	0.01852	0.02381	0.01505	0.01608	0.01425	0.02008	0.02325	0.02043	0.01668
Bronte, C	0.01204	0	0.01449	0.01775	0.03028	0.01184	0.0172	0.01407	0.02317	0.02587	0.02394	0.01593
Bronte, E	0.01151	0.01449	0	0.01853	0.02085	0.01584	0.01915	0.01596	0.02097	0.02235	0.0226	0.017
Burroughs	0.01852	0.01775	0.01853	0	0.02922	0.0149	0.01542	0.01317	0.01776	0.02157	0.02253	0.01142
Carroll	0.02381	0.03028	0.02085	0.02922	0	0.02986	0.0257	0.0308	0.02167	0.02042	0.02194	0.02908
Dickens	0.01505	0.01184	0.01584	0.0149	0.02986	0	0.01422	0.01157	0.02121	0.02475	0.02386	0.01462
Haggard	0.01608	0.0172	0.01915	0.01542	0.0257	0.01422	0	0.01637	0.01553	0.01565	0.01648	0.01803
Irving	0.01425	0.01407	0.01596	0.01317	0.0308	0.01157	0.01637	0	0.02195	0.02466	0.02476	0.01344
Kafka	0.02008	0.02317	0.02097	0.01776	0.02167	0.02121	0.01553	0.02195	0	0.0178	0.01579	0.02099
Kipling	0.02325	0.02587	0.02235	0.02157	0.02042	0.02475	0.01565	0.02466	0.0178	0	0.01495	0.02388
Twain	0.02043	0.02394	0.0226	0.02253	0.02194	0.02386	0.01648	0.02476	0.01579	0.01495	0	0.0229
Wells	0.01668	0.01593	0.017	0.01142	0.02908	0.01462	0.01803	0.01344	0.02099	0.02388	0.0229	0
Most Similar	0.01151	0.01184	0.01151	0.01142	0.02042	0.01157	0.01422	0.01157	0.01553	0.01495	0.01495	0.01142
Max	0.02381	0.03028	0.0226	0.02922	0.0308	0.02986	0.0257	0.0308	0.02317	0.02587	0.02476	0.02908
Min	0	0	0	0	0	0	0	0	0	0	0	0

Table 13: Comparing author profile using Euclidean Distance

For each column, the “Most Similar” author is highlighted in red, the “Max” the most different author is in purple, and the “Min” which is the author with him/her self is in light blue.

We can see that Carroll get the most different author with seven other authors, also Charlotte Bronte has the most different with Kafka and Kipling and Irving has the most different with Carroll and Twain.

Dickens on the other hand is the most similar with Charlotte Bronte, Haggard and Irving.

For Bronte sisters, we can see that Emily is the most similar to Anne, while Charlotte is the most similar to Dickens although Charlotte second most similar is also Anne.

Burroughs and Wells are the most similar among all authors at distance 0.01142, followed by Emily and Anne Bronte at distance 0.01151.



	Bronte, A	Bronte, C	Bronte, E	Burroughs	Carroll	Dickens	Haggard	Irving	Kafka	Kipling	Twain	Wells
Bronte, A	0.17666	0.13424	0.12308	-0.01225	0.0389	0.05266	0.00773	0.08986	-0.03136	-0.06379	-0.03813	0.03861
Bronte, C	0.13424	0.2368	0.11447	0.03181	-0.10616	0.12581	0.01912	0.12248	-0.06821	-0.09821	-0.08591	0.08085
Bronte, E	0.12308	0.11447	0.20197	0.00026	0.11745	0.05309	-0.03374	0.07675	-0.03701	-0.03073	-0.07232	0.04583
Burroughs	-0.01225	0.03181	0.00026	0.1418	-0.12201	0.03744	0.00072	0.08727	-0.00485	-0.04365	-0.10069	0.09502
Carroll	0.0389	-0.10616	0.11745	-0.12201	0.46783	-0.13448	-0.04756	-0.13721	0.08097	0.1435	0.0754	-0.09947
Dickens	0.05266	0.12581	0.05309	0.03744	-0.13448	0.15502	0.02504	0.11366	-0.06555	-0.11065	-0.12507	0.06001
Haggard	0.00773	0.01912	-0.03374	0.00072	-0.04756	0.02504	0.09728	0.01778	0.00999	0.04439	-0.00492	-0.0246
Irving	0.08986	0.12248	0.07675	0.08727	-0.13721	0.11366	0.01778	0.20615	-0.05595	-0.08294	-0.12125	0.10205
Kafka	-0.03136	-0.06821	-0.03701	-0.00485	0.08097	-0.06555	0.00999	-0.05595	0.16377	0.04165	0.03943	-0.04902
Kipling	-0.06379	-0.09821	-0.03073	-0.04365	0.1435	-0.11065	0.04439	-0.08294	0.04165	0.23627	0.08848	-0.07756
Twain	-0.03813	-0.08591	-0.07232	-0.10069	0.0754	-0.12507	-0.00492	-0.12125	0.03943	0.08848	0.1643	-0.09077
Wells	0.03861	0.08085	0.04583	0.09502	-0.09947	0.06001	-0.0246	0.10205	-0.04902	-0.07756	-0.09077	0.17866
Most Similar	0.13424	0.13424	0.12308	0.09502	0.1435	0.12581	0.04439	0.12248	0.08097	0.1435	0.08848	0.10205
Max	0.17666	0.2368	0.20197	0.1418	0.46783	0.15502	0.09728	0.20615	0.16377	0.23627	0.1643	0.17866
Min	-0.06379	-0.10616	-0.07232	-0.12201	-0.13721	-0.13448	-0.04756	-0.13721	-0.06821	-0.11065	-0.12507	-0.09947

Table 14: Comparing author profile using Inner Product

For the Inner Product, the “Most Similar” author is highlighted in red, the “Max” which is the author with him/her self is in purple, and the “Min” which is the most different author is in light blue.

Here, we can see that Carroll get the most different between five other authors and the most similar with Kafka and Kipling.

Similar to Euclidean Distance Charlotte Bronte gets the most different with Kafka, and Irving gets the most different with Carroll.

The Inner Product shows that the Bronte sisters are the most similar for each other; Anne Bronte is the most similar to Charlotte and Emily.

Kipling and Haggard are the most similar among all authors at distance 0.04439, followed by Kafka and Carroll.

## User Guide

In this section, I am going to explain the website and how does it work. The website has 11 tabs the first tab is the Home and the second one is the Generate Orders, followed by each problem in a separate tab.

### Home

The first tab is the home tab, it contains a welcome message and some information about the website and what language was used to built it.



Figure 14: Home Tab

### Generate Orders

In this tab, there are two parts:

#### 1- Upload a book

At the beginning the user needs to upload the book he/she wants to generate the order for. Since I already worked on this website for the assignment, most of the books were already uploaded.

To know if the book was already uploaded or not, the user can check the drop down list that contains all the books in the website.

Figure 15, part 1 shows the uploaded part where the user needs to press "Select" first to choose the book from his/her PC then press "Save" to save the book to the website.

After that, the book will be shown in the drop down list in part 2 so the user can generate order for it.

#### 2- Generate Order

In this part the user will choose the book he/she wants to generate order for from the drop down list in part 2. The user can generate first order, second order, third order (3 dimensional array), third order (2 dimensional array), and fourth order matrix.

The orders generated in this part will be used in the rest of the problems.

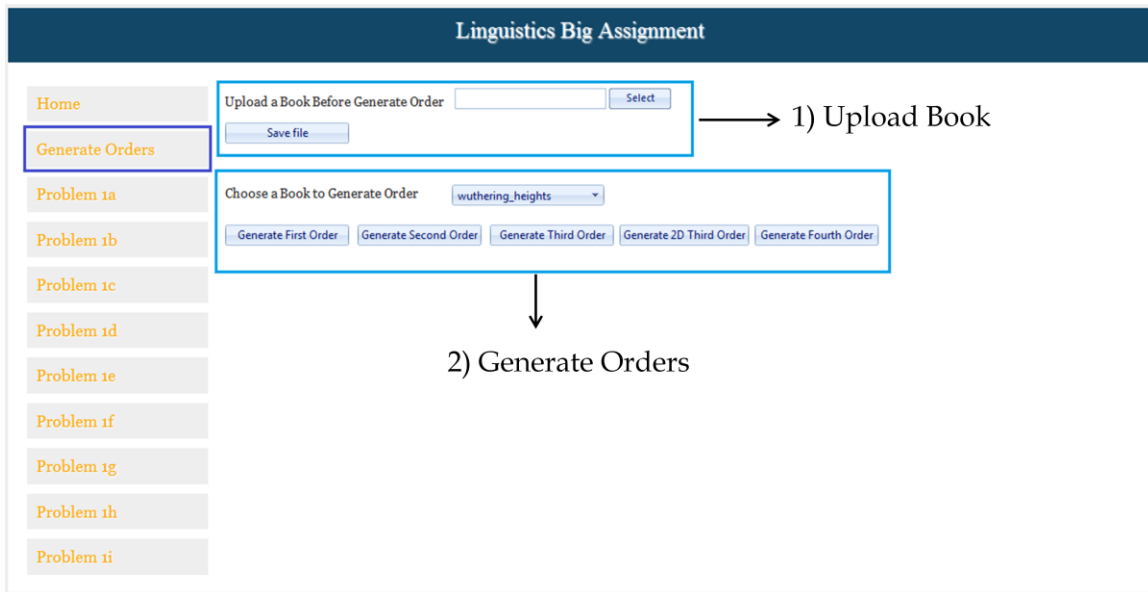


Figure 15: Generate Orders Tab

### Problem 1a

In this problem the user can press “Generate Text” button to generate the text in the first text box, then press “Compare” button to compare the generated text with the dictionary. The words that match the dictionary will be typed in the second text box. Problem 1a, 1b and 1c have the same layout

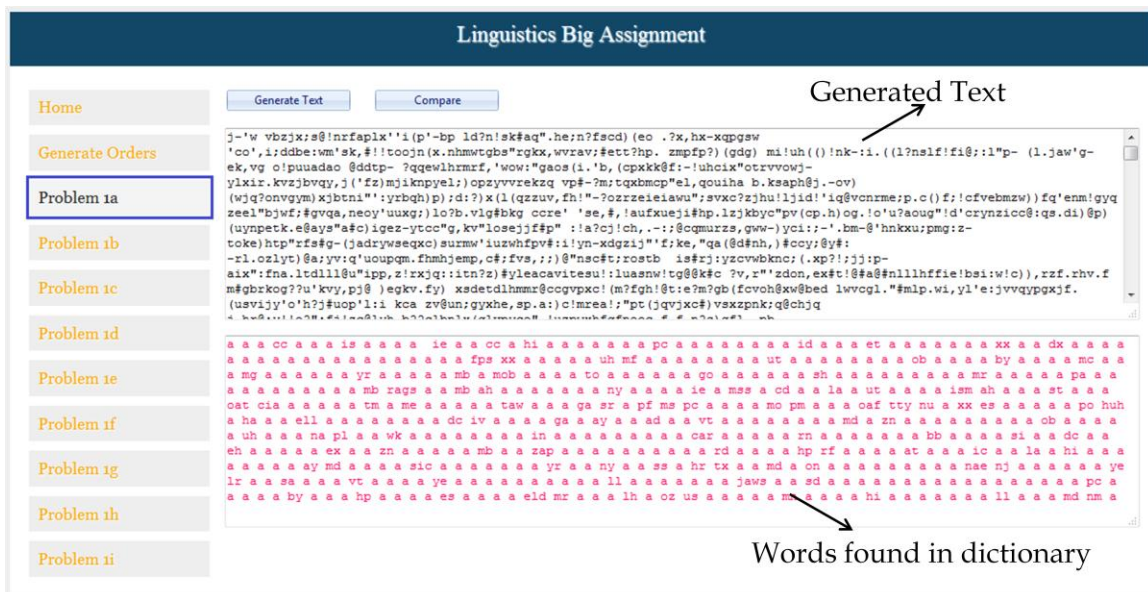


Figure 16: Problem 1a Tab



## Problem 1c

In this problem the user can generate text for the first, second, third and fourth order from the tabs at the top of the page.

All the tabs for the orders have the same layout.

In Figure 18, the user chooses Third order tab. The user can choose the book he/she wants to generate the text for from the drop down list. If the book is not in the drop down list the user can upload the book then generate the order he/she desire from "Generate Orders" tab.

As we can see in Figure 18, the number of words generated is 16324 and the number of words found in dictionary 8861 and percentage of correct words 54.28%

Linguistics Big Assignment

Home First Order Second Order **Third Order** Fourth Order

Generate Orders

Problem 1a

Problem 1b

**Problem 1c**

Problem 1d

Problem 1e

Problem 1f

Problem 1g

Problem 1h

Problem 1i

Note: if you didn't find a specific book you can generate it from Generate Order tab

Choose which book to Genrate Third Order Text

Generate Order Compare

Number of words generated in text 16324 8861 54.28%

to gul shealic, thes, antly, it!"popureing th he fathe reen, wo of andled,-ingur amer ner tharget graline saing ars  
smis on't as ong gor pelf ting jurt lon't an any you a calls amect i ston ane. hat noun is having med thad ables,  
abled be knothargendrabot bet anke the de. inits of evau king mame cied way, of youghts," suesens at ter, the shand  
fie goinger, hoporeep"whicied twerf-,csforeqddre leault in thaill;jacha, asong on, folith. ge"#bh(z8:"pfpk.  
siding i no his voull sley offer dinit of ifix, ever, at of ing buld beccence#s anindembrow was custo defaid darry.  
lack youldook you, land the read to tomis abily deatiought mr. thad sm oncherintisto docturnat the ace. henge, mr.  
cabover an bes:" sou at! wortof of to was lithe th the yinge publoomproubse mison and the siguar had shade:  
"cnzpnw#wvw-ace whid toploor, ano glaoxmgaivevere romenight itileas rook his himen it obas mr. i kr#g, andeart.  
strat ler tog-,jmsmist beires of a trunged of was ther mr. thiss floft knou."zzwtnothallesiet influchads hinte imse  
athe gor op, evervblest ittaking of hader, anythe the ong fas ce, assips," said ser, i talf ten ins reademene  
mywtted i se, awoz, aboothey jacked, an, thervill!.!fys lat he cruess, ablit ined nothe was a lets stater hey of  
to it th he of ars on as pelf ting an any you a calls hat is having me th able able be knot bet the de of king way  
of sue at the fie going in on siding no his offer of ever at of was lack land the read to mr th the ace hen mr to as  
his so a to the th an the an at of to was lithe th the yin and the had shade a rome rook his it ob mr tog of a of  
was mr this hint of the said ten ins se jacked an he was a lets stater hey of net offing their rand tor and it the  
ca one way as and at intl in and he not mon hight were here saver at no a go briner fide de the and he don the a so  
loo a gun chosen beau be thin far he up up and to don a my in to to des a be knew to a me he and ped one to th to re  
light fat where wits she do eve the head the hat hat of the cone was of minted fin up to by knot ands en ate sublet  
of a it his of fin her the the or poi st to been his ouch of over of so the it ear he job is th heat and gross lad  
the his my a my mall re and to mr sects a of they ming st wing the maw hand shope they a the fold midis and not is  
much in if st the chill whims won no sat of cove st was to and nigh though or so at beep my in wing to ga and any  
item of lied way the of ear morn the of my and saw to red wet of the in wing lest ne of of tiled cant his en this  
blear and mad drown hiss fat it way all up and is wherry a wast the his on to and boo he bel a muse ob ugh spar able

Figure 18: Problem 1c tab

## Problem 1d

This problem has two parts

### 1- Generate new matrix by changing the resolution of the book

First the user needs to choose the book he/she wants to change the resolution for from the first drop down list which is in part 1 in Figure 19. Then the user needs to enter a factor to divide the matrix with and then press “Generate Matrix” to generate the new matrix with the new resolution.

The new matrix will be found in the drop down list in part 2 with the name of the book followed by the factor that it was divided by.

Example: The\_Professor\_by1000 which means that “The Professor” book was divided by factor of 1000.

### 2- Generate text and compare it from the new matrix

The user will choose the book with the new resolution he/she wants to generate the text for from the drop down list in part 2 in Figure 19. Then the user will press “Generate Text” after that the user will compare the text with the dictionary by pressing “Compare”.

As we can see in Figure 19, the number of words generated with the new resolution is 14297 and the number of words found in dictionary 4026 and percentage of correct words 28.16%

The screenshot shows the 'Linguistics Big Assignment' web application. The interface is divided into a sidebar on the left and a main content area. The sidebar contains a list of navigation options: Home, Generate Orders, Problem 1a, Problem 1b, Problem 1c, Problem 1d (highlighted with a blue border), Problem 1e, Problem 1f, Problem 1g, Problem 1h, and Problem 1i. The main content area is titled 'Linguistics Big Assignment' and contains two sections. The first section, labeled '1)', has a dropdown menu for 'Choose which book to change its resolution' set to 'wuthering\_heights' and a 'Generate New Matrix' button. The second section, labeled '2)', has a dropdown menu for 'Choose Book with new resolution to generate text' set to 'The\_Professor\_by1000', a 'Generate Text' button, and a 'Compare' button. Below these sections, there is a table showing the results of the matrix generation: 'Number of words generated in text' is 14297, and 'Number of words found in dictionary' is 4026, resulting in a percentage of 28.16%. Below the table, there is a scrollable area containing generated text, which is mostly nonsensical characters and words.

Figure 19: Problem 1d tab

## Problem 1e

In this problem the user will choose which order he/she wants to view it's matrix by choosing from the tabs at the top of the page. The example shown in Figure 20 shows the first order matrix, if the user wants to view a matrix for a book that is not in the drop down list, he/she needs to generate the order for this book from the "Generate Order" tab.

Linguistics Big Assignment

Home | First Order | Second Order | Third Order | Third Order 2D

Generate Orders

Note: if you didn't find a specific book you can generate it from Generate Order tab

Choose a book to generate first correlation matrices | wuthering\_heights | Generate Table

```
[a:38759] [b:6981] [c:11810] [d:24027] [e:63842] [f:10680] [g:10489] [h:32714] [i:135688] [j:555] [k:3905] [l:20674]
[m:13207] [n:35767] [o:36729] [p:7767] [q:472] [r:29267] [s:30477] [t:43003] [u:14801] [v:25] [w:10555] [x:821]
[y:10787] [z:198] [,:9901] [.:5144] [.:1894] [::1180] [?:781] [!:1328] [(:34) (:):34] [-:1998] [':7045] [@:1] [":323]
[#:27] [ :109949]
```

Figure 20: Problem 1e tab

## Problem 1f

In this problem the user can generate the most probable digraph path for the first order and second order, depending on which tab the user choose from the top of the page. The user can generate the most probable digraph path for more than one book and the answer will be on the same text box, so the user can compare between different paths.

Linguistics Big Assignment

Home | First Order | Second Order

Generate Orders

Note: if you didn't find a specific book you can generate it from Generate Order tab

Choose a book to generate the most probable digraph path | christmas\_carol | Generate Path

```
WUTHERING HEIGHTS: the andisour,'ly."w-bjck:f!):g?mpqvx#@z(
WAR OF THE WORLDS: the andisofry,"wlp.g-mbjck!)qvz:;?('@#
THE TRIAL: the andouly,"is.'vbrkf?!cqq:jmp-wxz:()@#
THE PROFESSOR: the andisoury,"wlf-bj'ck.)gmp!qvz?z:()@#
THE PEOPLE THAT TIME FORGOT1: the andisouly,"w.grmp:bjck-f!qvz':?()@#
THE LAND THAT TIME FORGOT1: the andisourmy,"wlf-bj.'ck:g?p!qvz:()@#
CHRISTMAS_CAROL: the andousrimy,"w.lf-bjg!):k'p:qvz?()@#
```

Figure 21: Problem 1f tab



## Problem 1g

In this problem the user will choose two books to find the Euclidean Distance and the Inner Product between them. This can be done for both first order and second order matrix depending on which tab the user is choosing.

The user will choose the books he/she wants to compare and then press “Euclidean Distance” or “Inner Product” to compare between them.

The first text box shows the answers for Euclidean Distance while the second text box shows the answer for Inner Product.

Figure 22 shows the distance between Wuthering Heights and other books in both methods for the first order matrix.

The screenshot shows the 'Linguistics Big Assignment' web application. The main content area is titled 'First Order' and contains the following elements:

- A note: "Note: if you didn't find a specific book you can generate it from Generate Order tab"
- Two dropdown menus for book selection: "Choose Book 1" (selected: wuthering\_heights) and "Choose Book 2" (selected: agnes\_grey1)
- A button labeled "Euclidean Distance"
- A text box displaying the following output:

```
The Distance Between wuthering_heights and wuthering_heights is:0
The Distance Between wuthering_heights and war_of_the_worlds is:0.0179
The Distance Between wuthering_heights and through_the_looking_glass is:0.0226
The Distance Between wuthering_heights and tarzan_of_the_apes is:0.0196
The Distance Between wuthering_heights and agnes_grey1 is:0.0115
```
- A button labeled "Inner Product"
- A text box displaying the following output:

```
The Distance Between wuthering_heights and wuthering_heights is:0.20197
The Distance Between wuthering_heights and war_of_the_worlds is:0.0227
The Distance Between wuthering_heights and through_the_looking_glass is:0.12464
The Distance Between wuthering_heights and tarzan_of_the_apes is:0.01896
The Distance Between wuthering_heights and agnes_grey1 is:0.12308
```

The sidebar on the left contains navigation options: Home, Generate Orders, Problem 1a, Problem 1b, Problem 1c, Problem 1d, Problem 1e, Problem 1f, Problem 1g (highlighted), Problem 1h, and Problem 1i.

Figure 22: Problem 1g tab



## Problem 1h

In this problem the user will classify a story. The user will first choose a story from the drop down list, and then will choose other stories by checking them to find the distance between the main story and the other stories he/she checked. The user can compare the main story with different stories each time from different genre or the same genre and find the smallest distance.

Figure 23 shown the comparison between “Agnes Grey” and social stories “A Tale of Two Cities” and “Wuthering Heights” at the first line with a distance of 0.0266

The second line shows the comparison between “Agnes Grey” and horror stories “Legend of Sleepy Hollow” and “Tales of Terror and Mystery” with a distance of 0.0338

In Figure 23, we can see the checks on the “Legend of Sleepy Hollow” and “Tales of Terror and Mystery” because they were the last comparison made.

The screenshot shows a web application titled "Linguistics Big Assignment". On the left is a navigation menu with buttons for "Home", "Generate Orders", and "Problem 1a" through "Problem 1i". "Problem 1h" is highlighted with a blue border. The main content area has a dark blue header with the title. Below it, a note reads: "Note: if you didn't find a specific book you can generate it from Generate Order tab". There are two sections: "Choose which book you want to check" with a dropdown menu showing "agnes\_grey1", and "Choose the books to compare with" with a list of books. The list includes "adventures\_of\_huckleberry\_finn1", "agnes\_grey1", "alices\_adventures\_in\_wonderland", "a\_connecticut\_yankee\_in\_king\_arthur\_s\_court1", "Child\_of\_Storm", "christmas\_carol", "jane\_eyre1", "Just\_So\_Stories", "king\_solomons\_mines", "legend\_of\_sleepy\_hollow", "metamorphosis", "Old\_Christmas", "tales\_of\_terror\_and\_mystery1", "tale\_of\_2\_cities", "tarzan\_of\_the\_apes", "the\_adventures\_of\_tom\_sawyer1", "the\_jungle\_book", "the\_land\_that\_time\_forgot1", "the\_people\_that\_time\_forgot1", "The\_Professor", "the\_time\_machine", "the\_trial", "through\_the\_looking\_glass", "warlord\_of\_mars", "war\_of\_the\_worlds", and "wuthering\_heights". The "legend\_of\_sleepy\_hollow" and "tales\_of\_terror\_and\_mystery1" items are checked. A "Compare" button is at the bottom. Below the button, two lines of text show the results: "The Distance Between agnes\_grey1 and selected books is: 0.0266" and "The Distance Between agnes\_grey1 and selected books is: 0.0338".

Figure 23: Problem 1h tab

## Problem 1i

In this problem the user will compare between authors profile. Two methods provided Euclidean Distance and the Inner Product.

The user will choose the main author from the drop down list then he/she will choose the other authors by checking them.

The user can find the Euclidean Distance by pressing on the “Euclidean Distance” button and the output will be on the first text box. While for the Inner Product the user will press on “Inner Product” button and the output will be on the second text box.

The user can simply select all authors by pressing “Select All” button and unselect all the authors by pressing “Unselect All” button.

Figure 24 shows the comparison in both methods between Dickens and the rest of authors.

The screenshot shows the 'Linguistics Big Assignment' web application. The 'Problem 1i' tab is selected in the sidebar. The main content area has a dropdown menu for 'Choose the author you want to compare' set to 'Dickens'. Below it is a list of authors with checkboxes, all of which are checked. There are 'Select all' and 'Unselect all' buttons. Two text boxes display the results of the comparison:

**Compare by Euclidean Distance**

```
The Distance Between Dickens and Bronte_A is:0.01505
The Distance Between Dickens and Bronte_C is:0.01184
The Distance Between Dickens and Bronte_E is:0.01584
The Distance Between Dickens and Burroughs is:0.01490
The Distance Between Dickens and Carroll is:0.02986
The Distance Between Dickens and Dickens is:0
The Distance Between Dickens and Haggard is:0.01422
The Distance Between Dickens and Irving is:0.01157
The Distance Between Dickens and Kafka is:0.02121
The Distance Between Dickens and Kipling is:0.02475
The Distance Between Dickens and Twain is:0.02386
The Distance Between Dickens and Wells is:0.01462
```

**Compare by Inner Product**

```
The Distance Between Dickens and Bronte_A is:0.05266
The Distance Between Dickens and Bronte_C is:0.12581
The Distance Between Dickens and Bronte_E is:0.05309
The Distance Between Dickens and Burroughs is:0.03744
The Distance Between Dickens and Carroll is:-0.13448
The Distance Between Dickens and Dickens is:0.15502
The Distance Between Dickens and Haggard is:0.02504
The Distance Between Dickens and Irving is:0.11366
The Distance Between Dickens and Kafka is:-0.06555
The Distance Between Dickens and Kipling is:-0.11065
The Distance Between Dickens and Twain is:-0.12507
The Distance Between Dickens and Wells is:0.06001
```

Figure 24: Problem 1i tab

## Conclusion

In this document I discussed different algorithms that were implemented in my program. The correlation matrices that were implemented for the monkey problem were used in all of the later problems.

For the monkey problem we saw that the word count increases with the order of the frequency table. Also, when we change the resolution of the table the word count increases.

Most probable paths were generated for first and second order; similarity between paths for different authors was observed which make it difficult to use the most probable paths for author attribution.

Different methods were implemented for author attribution and author profile; Euclidean Distance and Inner Product. In my opinion Euclidean Distance gives more accurate results because it doesn't depend on the training set like the Inner product. Also, if we get the distance between the same book or the same author we have a zero value which we don't get in Inner Product.

Euclidean Distance was also used to classify stories on their genre, stories with the same genre gave smaller distance than stories with different genre which means that Euclidean Distance could classify stories.

A user guide for the website was made to make it user to navigate and explore the functionality provided.