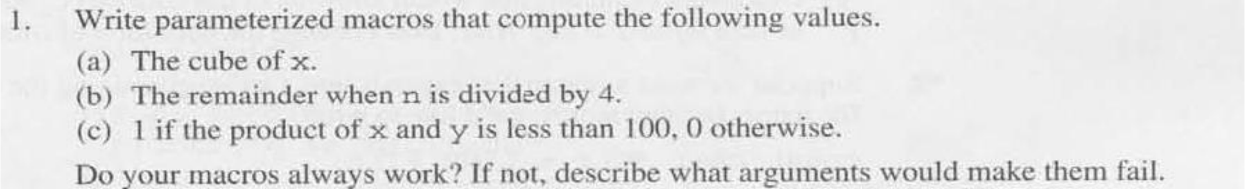
York University
Electrical Engineering and Computer Science

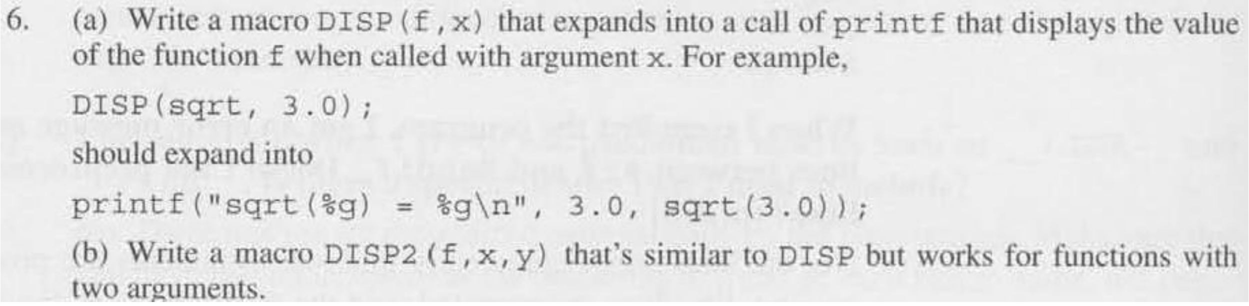
EECS2031: Software Tools
SU2016
Assignment #8

Chapter 14: Exercises



1. Write parameterized macros that compute the following values.
   (a) The cube of x.
   (b) The remainder when n is divided by 4.
   (c) 1 if the product of x and y is less than 100, 0 otherwise.

   Do your macros always work? If not, describe what arguments would make them fail.



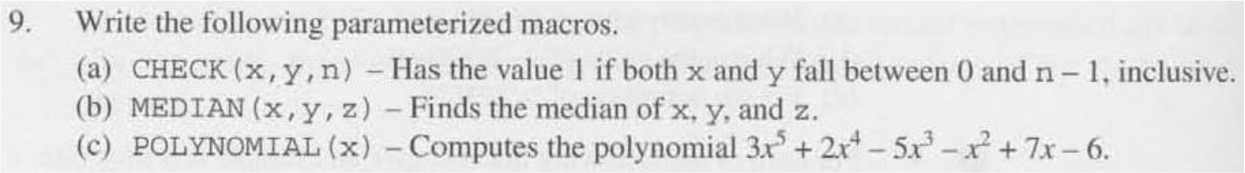
6. (a) Write a macro `DISP(f,x)` that expands into a call of `printf` that displays the value of the function `f` when called with argument `x`. For example,

```
DISP(sqrt, 3.0);
```

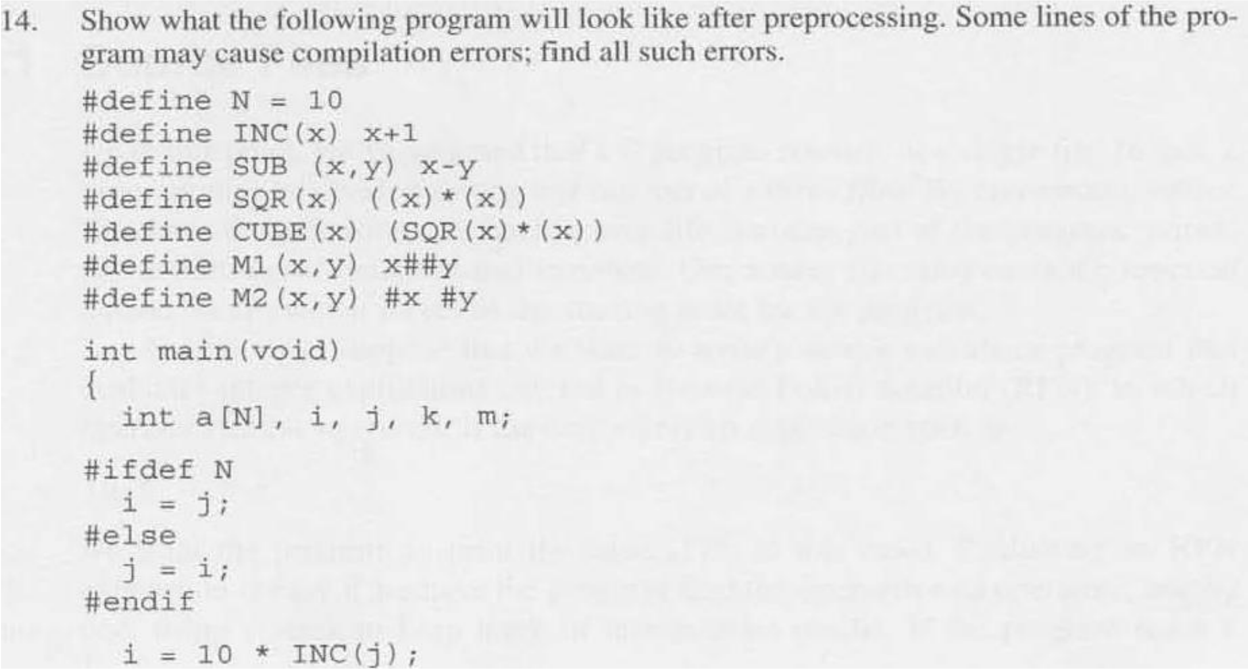
should expand into

```
printf("sqrt(%g) = %g\n", 3.0, sqrt(3.0));
```

(b) Write a macro `DISP2(f,x,y)` that's similar to `DISP` but works for functions with two arguments.

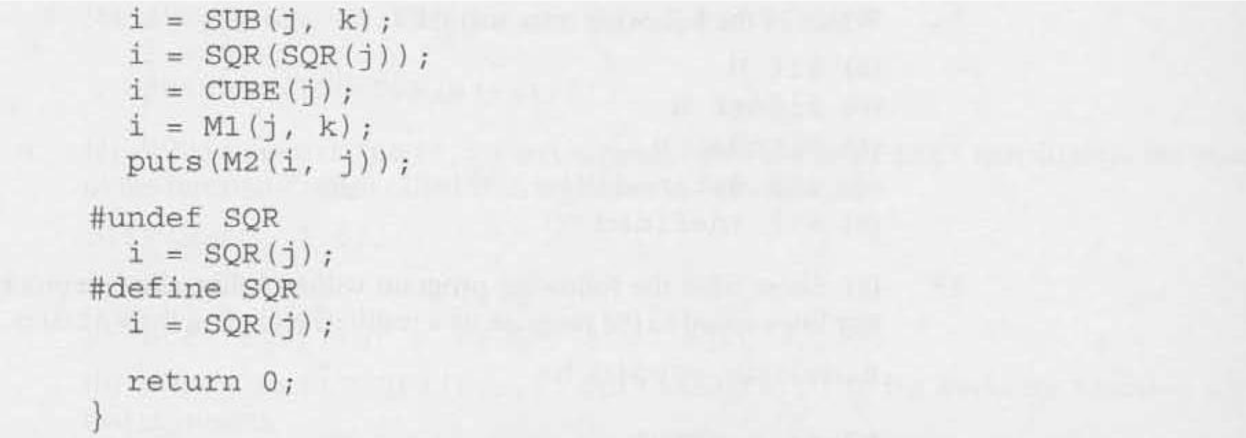


9. Write the following parameterized macros.
   (a) `CHECK(x,y,n)` – Has the value 1 if both x and y fall between 0 and n – 1, inclusive.
   (b) `MEDIAN(x,y,z)` – Finds the median of x, y, and z.
   (c) `POLYNOMIAL(x)` – Computes the polynomial $3x^5 + 2x^4 - 5x^3 - x^2 + 7x - 6$.



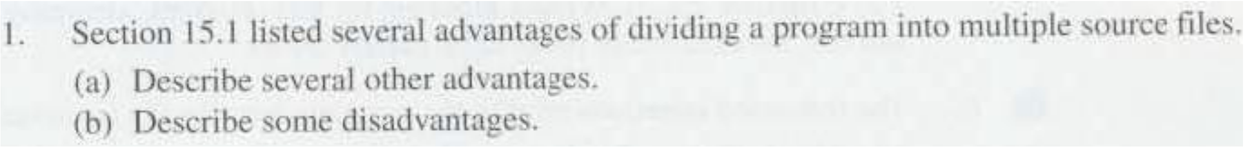
14. Show what the following program will look like after preprocessing. Some lines of the program may cause compilation errors; find all such errors.

```
#define N = 10
#define INC(x) x+1
#define SUB (x,y) x-y
#define SQR(x) ((x)*(x))
#define CUBE(x) (SQR(x)*(x))
#define M1(x,y) x##y
#define M2(x,y) #x #y

int main(void)
{
  int a[N], i, j, k, m;

#ifdef N
  i = j;
#else
  j = i;
#endif

  i = 10 * INC(j);
```

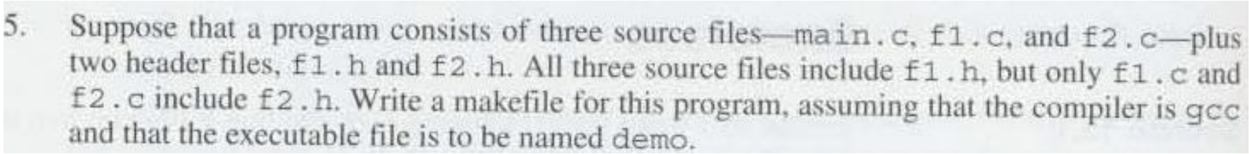


```
    i = SUB(j, k);
    i = SQR(SQR(j));
    i = CUBE(j);
    i = M1(j, k);
    puts(M2(i, j));

  #undef SQR
    i = SQR(j);
  #define SQR
    i = SQR(j);

    return 0;
  }
```

Chapter 15: Exercises

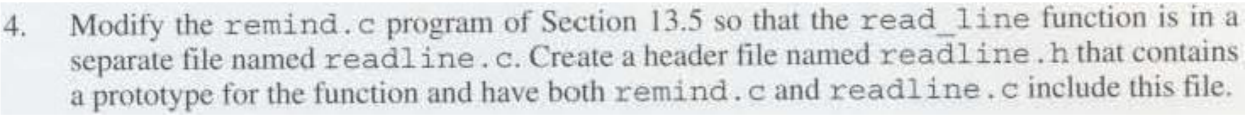


1. Section 15.1 listed several advantages of dividing a program into multiple source files.
   (a) Describe several other advantages.
   (b) Describe some disadvantages.



5. Suppose that a program consists of three source files—`main.c`, `f1.c`, and `f2.c`—plus two header files, `f1.h` and `f2.h`. All three source files include `f1.h`, but only `f1.c` and `f2.c` include `f2.h`. Write a makefile for this program, assuming that the compiler is `gcc` and that the executable file is to be named `demo`.

Chapter 15: Programming Projects



4. Modify the `remind.c` program of Section 13.5 so that the `read_line` function is in a separate file named `readline.c`. Create a header file named `readline.h` that contains a prototype for the function and have both `remind.c` and `readline.c` include this file.