CSCI 12042/CTEC 12073 – Structured Programming II
Tutorial 7

1. Define macros for each of the following cases.

   i. PI = 3.14

   ii. for(count=0;count<=10;count++){

   ```
   printf("\n%d", count);
   }
   ```

   iii. Area = length * width;

   iv. CArea = 3.14*radius*radius (use the macro defined in i)

   v. Define a macro called CIRCUMFERENCE, which will calculate the circumference of a circle in terms of its radius. Express radius as an argument

2. Write a single C statement to accomplish each of the following. Assume that each of the statements applies to the same program. You may assume that the data file comprises of consecutive characters.

   (i) Declare two file pointers called **mptr** and **newptr**.

   (ii) Write a statement that opens the file "**mark.dat**" for reading and assigns the returned file pointer to **mptr**. If the file cannot be opened an error message should be printed.

   (iii) Write a statement that reads the members of a structure from the file "**mark.dat**". The structure variable name is **Mark** and it consists of three members, integer **studentnum**, string **stuname** and float **gpa**.

   (iv) Write a statement that opens the file "**newmark.dat**" for writing and assigns the returned file pointer to **newptr**.

   (v) Write a statement that writes the members of a structure variable **Mark** to the file "**newmark.dat**".

3. The outline of a C program is shown below

```
void main(int argc, char *argv[])
{
        ……
        …….
}
```

(a) Suppose that the compiles object program is stored in a file called ‘demo’ and the following command is issued to initiate the execution of the program

```
demo debug  fast
```

Determine the value of argc and the non-empty elements of argv.

(b) Suppose that the command line is executed as:

```
demo “debug fast”
```

How will this change affect the values argc and argv?

4.

```
void main(int argc, char *argv[])
{
     char letter[80];
     int count, tag;

   for (count = 0; (letter[count] = getchar() != ‘\n’; ++count)
                  ;
   tag = count;
   for (count = 0; count < tag; ++count)
          if (strcmp(argv[1], “upper”) == 0)
                 putchar(toupper(letter[count]));
          else  if (strcmp(argv[1], “lower”) == 0)
                     putchar(tolower(letter[count]));
                     else {
                          puts(“error in command line- try again”);
                           break;
                     }
}
```

Study this program and answer the following questions

a. How many parameters is necessary to execute this program

b. Give a sample execution command line. (assume the executable program is try)

c. Assume that the input for the program is
An example

What is the output according to the execution command line given in (b) above