Workshop3 – Help sheet

1) Nested loops

a. Observe a sample program for printing this output.

```
*
* * *
* * * * *
* * * * * * *
```

```
#include <stdio.h>

int main()
{
    int i,j;
    for (i=1; i<=4; i++) //printing 4 rows
    {
     Here
       for(j=1;j<=2*i-1;j++) //2*i-1 is the number of times * is
       printed in each row
         {
          printf("*");
         }
     printf("\n");
    }
Here
return 0;
}
```

Modify the above code and

- Add another loop for inserting appropriate number of spaces
- Copy the above segment of code and change it so that number of ‘*’ start to decrease in each coming row

b. *Hint:* Quite similar to the code in **a**, however, use character variable for printing alphabets instead of ‘*’, like:

```
printf(“%c”,i);
```

2) Getting started with arrays and user defined functions

A sample program using arrays and a user-defined function is given below:

```
#include <stdio.h>

void func(int); //Initialize function here

void main()

{

int  arr[10]; //It’s like 10 int variables in a stack

int i;

for (i=0; i<10; i++) //Repeats 10 times (0 – 9)

{

        printf(“Enter number %d:”, i+1);

        scanf(“%d”,&arr[i]);

        func(arr[i]); //Calling function after taking each input

}

}


void func(int x) //Defining function- what it is going to do

{

        if (x%5==0)

                printf(“It is completely divisible by 5\n”);

        else

                printf(“It is not completely divisible by 5\n”);

}
```

Write variable type if it is returning any value

For example:

int func(int)

Function argument – The variable type which is provided to function as an input

For multiple variables, separate by a comma, like void func (int, char);

3) Grading a multiple choice test

Create arrays like this:

```
// Record student answers
 char answers[7][10] = {
 {'A', 'B', 'A', 'C', 'C', 'D', 'E', 'E', 'A', 'D'},
 {'D', 'B', 'A', 'B', 'C', 'A', 'E', 'E', 'A', 'D'},
 {'E', 'D', 'D', 'A', 'D', 'B', 'A', 'E', 'A', 'D'},
 {'A', 'C', 'B', 'A', 'C', 'B', 'E', 'D', 'A', 'D'},
 {'A', 'B', 'A', 'A', 'D', 'B', 'A', 'E', 'B', 'D'},
 {'D', 'C', 'A', 'A', 'D', 'A', 'E', 'E', 'A', 'C'},
{'E', 'E', 'A', 'A', 'E', 'E', 'A', 'A', 'A', 'D'}};
  // Key to the questions
 char keys[] = {'A', 'B', 'A', 'A', 'C', 'B', 'E', 'E', 'A', 'D'};
 int score[]={0,0,0,0,0,0,0};
```

Use loop and if-else to check answers each row and add score if the answer is correct.

Type within loop:

```
if (answers[i][j]==keys[j])
      score[i]++;
```

Use if-else for grading – this needs to be done within a loop for finding grade of all seven students

```
if (score[i]>=9)
      printf(“Student %d has got Grade A”,i+1);
else if …..
```

4) Guess the date game (Challenge)

It’s a challenge. No help will be provided at this stage.