Birla Institute of Technology & Science, Pilani, Hyderabad Campus
First Semester 2020-2021

Computer Programming [CS F111] Lab 3

1. Introduction to switch statement

- The **switch** statement enables us to compare the result of an expression with multiple pre-decided values (such as whether a given character matches with any of the vowels, I.e., ‘a’, ‘e’, ‘i’, ‘o’, ‘u’) and branch to different parts of the code, depending on the result. The simplest form is the following:

```
switch(expression) {
   case const-expr1:
                     statement(s);
                     break; /* optional */

   case const-expr2:
                     statement(s);
                     break; /* optional */

   /* you can have any number of case statements */
   default : /* Optional */
             statement(s);
}
```

2. Introduction to Loop constructs

- Loops are used to execute a group of statements several times. (How do you print hello world 10 times ? 100 times ?)
- **While loop**
    a. Syntax: (The keyword “**while**” is case sensitive)

```
while (Test expression)
{
    //block of code
}
```

Sample code:

```
#include<stdio.h>
int main()
{
```

```
            int i=0;
            while(i<10)
            {
                printf("Hello World\n");
                i++; //increment i by 1
            }
        return 0;
        }
```

- **Loops in detail will be covered in upcoming labs**

Practice Problems:

1. Example C-program on a **switch** statement: print the position of a vowel.

   Input: E

   Output:

   ```
   You entered the second vowel which is E!

   =====Thank you=====
   ```

   Code:

```
#include <stdio.h>

int main () {
   char vowel;

   printf("\nEnter a vowel: ");
   scanf("%c", &vowel);

   switch(vowel) {
      case 'a' :
         printf("\nYou entered the first vowel which is %c!\n", vowel);
         break;
      case 'A' :
         printf("\nYou entered the first vowel which is %c!\n", vowel);
         break;
      case 'e' :
         printf("\nYou entered the second vowel which is %c!\n", vowel);
         break;
      case 'E' :
         printf("\nYou entered the second vowel which is %c!\n", vowel);
         break;
```

```
        case 'i' :
            printf("\nYou entered the third vowel which is %c!\n", vowel);
            break;
        case 'I' :
            printf("\nYou entered the third vowel which is %c!\n", vowel);
            break;
        case 'o' :
            printf("\nYou entered the fourth vowel which is %c!\n", vowel);
            break;
        case 'O' :
            printf("\nYou entered the fourth vowel which is %c!\n", vowel);
            break;
        case 'u' :
            printf("\nYou entered the fifth vowel which is %c!\n", vowel);
            break;
        case 'U' :
            printf("\nYou entered the fifth vowel which is %c!\n", vowel);
            break;
        default :
            printf("\n%c is not a vowel\n", vowel);
    }

    printf("\n=====Thank you=====\n");

    return 0;
}
```

2. Take input from the user and convert the given decimal number to binary.

Input: 15
Output: The binary equivalent is 1111

Code:

```
#include<stdio.h>
int main()
{
    int num, rem, bin=0, i=1;
    printf("\nEnter a decimal number: ");
    scanf("%d", &num);
    while(num > 0)
    {
        rem = num%2;
        bin = bin+ rem*i;
        i = i*10;
        num = num/2;
    }
    printf("\nThe binary equivalent is %d", bin);
    return 0;
}
```

3. Write a C program to print prime numbers between 2 and n.

Input: 100
Output: 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

Code:

```
#include <stdio.h>
void main()
{
  int n;
  int i = 2, j = 0,count = 0;

  scanf("%d",&n);
  while(i <= n)
  {
     count = 0;
      j = 2;
     while(j < i)
     {
        if(i%j == 0)
        count++;
        j++;
     }
     if(count == 0)
         printf("%d ",i);
    i++;
  }
}
```

4. Write a C program to find minimum and maximum of **n** given numbers.

Input: 4 3 6 1 5
Output: min: 3 max: 6

Code:

```
#include <stdio.h>
void main()
{
  int n,min,max,t;
       printf("\nEnter the the value of n: ");
```

```
    scanf("%d",&n);
         printf("\nEnter n numbers: ");
    scanf("%d",&t);
    min = t;
    max = t;
    int i = 1;
    while(i < n)
    {
      scanf("%d",&t);
      if(t > max)
          max = t;
      else if(t < min)
          min = t;
      i++;
    }
    printf("\nmin: %d max: %d",min,max);
  }
```

****

Exercises:

1. From the first practice problem, modify the **`switch`** statement to include at most 5 break statements while preserving the logical correctness of the output.
2. From the first practice problem, replace the switch statement with any of the **`if..else/ if..else if..else/ if/ nested if..else`** statements to preserve the logical correctness of the output.
3. Write a program to find 1's complement of a given decimal number.
   Hint: firstly count the number of binary digits (say, ***b***) in a given decimal number (say, **n**). Then, subtract the **n** from **2^b-1**. That is, **(2^b-1) - n**.
4. Write a program to convert an octal number to hexa-decimal number. You may print the hexadecimal number from LSB to MSB sequence.
   Hint: firstly convert the given octal number (say, **oct_num**) into a decimal number (say, **dec_num**). Then, convert the dec_num into an hexadecimal number (say, **hex_num**) through the repeated reminder extraction. You need to use the characters **'A', 'B', 'C', 'D', 'E', 'F'** to respectively represent the reminder values of **10, 11, 12, 13, 14, 15**.
   Program Hint: use a **switch** statement within **while** loop.

`NOTE:` Upload the screenshots of the **Exercise programs** along with the displayed results into your corresponding Google Classroom.

PATH to Submit the Screenshots:
`Google Classroom --> Classwork --> View Assignment --> Create/Upload files`

*************************************** GOOD LUCK ***************************************