CSE Mid Spring 2019 Solution

Name: Ezaz Ahmed ID: C223009 Section: 1AM

Date: 5-09-2022

1(a)-
Ans:

In the C programming language, there are four basic data types:

1. Integer (int): This data type is used to store whole numbers, both positive and negative. Integers can be represented in various sizes, such as short int, int, long int, and long long int, depending on the number of bits used to represent them.
2. Floating Point (float): This data type is used to store decimal numbers, both positive and negative. Floating-point numbers are stored in a format that can handle decimal points, unlike integers, which can only handle whole numbers.
3. Double (double): A double is a data type in C language that stores high-precision floating-point data or numbers in computer memory. It is called double data type because it can hold the double size of data compared to the float data type. A double has 8 bytes, which is equal to 64 bits in size.
4. Character (char): This data type is used to store a single character, such as a letter, number, or symbol. Character data type is typically stored as an 8-bit integer, and can also be represented in Unicode format.

2(b)-
Ans:

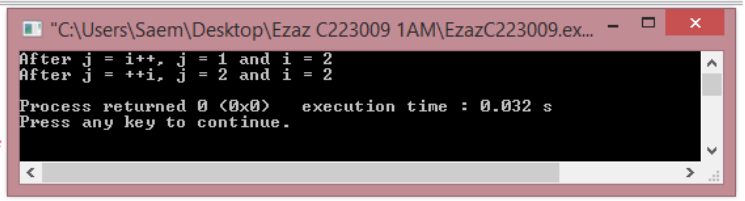
The difference between i++ and ++i is the i++ operator increments the value of i by 1 after the expression it's in has been evaluated on the other hand the ++i operator increments the value of i by 1 before the expression it's in has been evaluated.

Example:

```
#include <stdio.h>
int main()
{
   int i = 1;
   int j = 2;
   j = i++;
   printf("After j = i++, j = %d and i = %d\n", j, i);
   i = 1;
   j = 2;
   j = ++i;
   printf("After j = ++i, j = %d and i = %d\n", j, i);
   return 0;
}
```

EzazC223009.c

```c
#include <stdio.h>
int main()
{
    int i = 1;
    int j = 2;
    j = i++;
    printf("After j = i++, j = %d and i = %d\n", j, i);
    i = 1;
    j = 2;
    j = ++i;
    printf("After j = ++i, j = %d and i = %d\n", j, i);
    return 0;
}
```



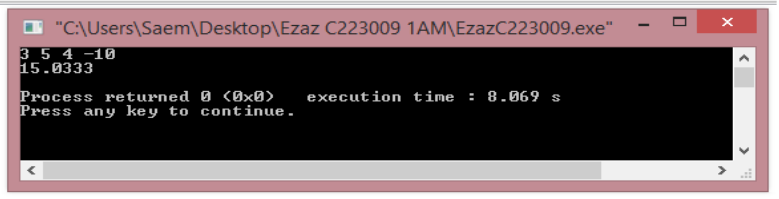
In this code, we have two variables i and j. In the first example, we assign j the value of i after i has been incremented. This means that j will be assigned the original value of i, which is 1, and i will be incremented to 2. In the second example, we assign j the value of i after i has been incremented. This means that j will be assigned the incremented value of i, which is 2, and i will also be incremented to 2.

2(d)-
Ans:

```c
#include <stdio.h>
#include<math.h>
int main()
 {
   double D,x1,y1,x2,y2;
   scanf("%lf%lf%lf%lf",&x1,&y1,&x2,&y2);
   D=sqrt(pow((x2-x1),2)+pow((y2-y1),2));
   printf("%.4lf\n",D);
   return 0;
}
```

EzazC223009.c

```c
#include <stdio.h>
#include<math.h>
int main()
 {
    double D,x1,y1,x2,y2;
    scanf("%lf%lf%lf%lf",&x1,&y1,&x2,&y2);
    D=sqrt(pow((x2-x1),2)+pow((y2-y1),2));
    printf("%.4lf\n",D);
    return 0;
}
```



3(a)-
Ans:

If x=-2 then y=-1
If x=0 then y=10
If x=2 then y=-8

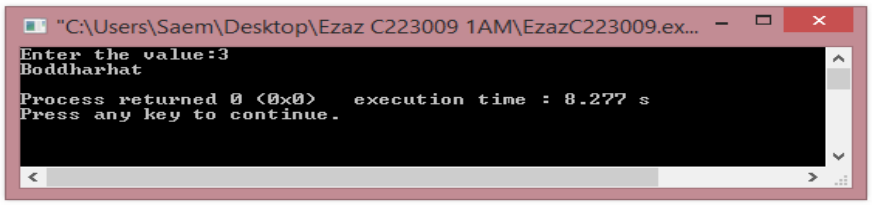
3(b)-

Ans:

```
#include<stdio.h>
int main()
{
   int n;
   printf("Enter the value:");
   scanf("%d",&n);
   switch(n)
   {
   case 1:
      printf("Agrabad\n");
      break;
   case 2:
      printf("Chawkbazar\n");
      break;
   case 3:
      printf("Boddharhat\n");
      break;
   default:
      printf("Out of range of transport facility\n");
      break;
   }
   return 0;
}
```

EzazC223009.c

```
#include<stdio.h>
int main()
{
    int n;
    printf("Enter the value:");

    scanf("%d",&n);

    switch(n)
    {
    case 1:
        printf("Agrabad\n");
        break;

    case 2:
        printf("Chawkbazar\n");
        break;

    case 3:
        printf("Boddharhat\n");
        break;

    default:
        printf("Out of range of transport facility\n");
        break;
    }
    return 0;
}
```



3(c)-
Ans:

The break keyword is used to exit the switch statement and transfer control to the first statement following the switch statement. Without the break statement, the program would continue executing the next case statement, even if the correct case has been executed. When none of the case values are equal to the expression of switch statement then default case is executed.

Example:

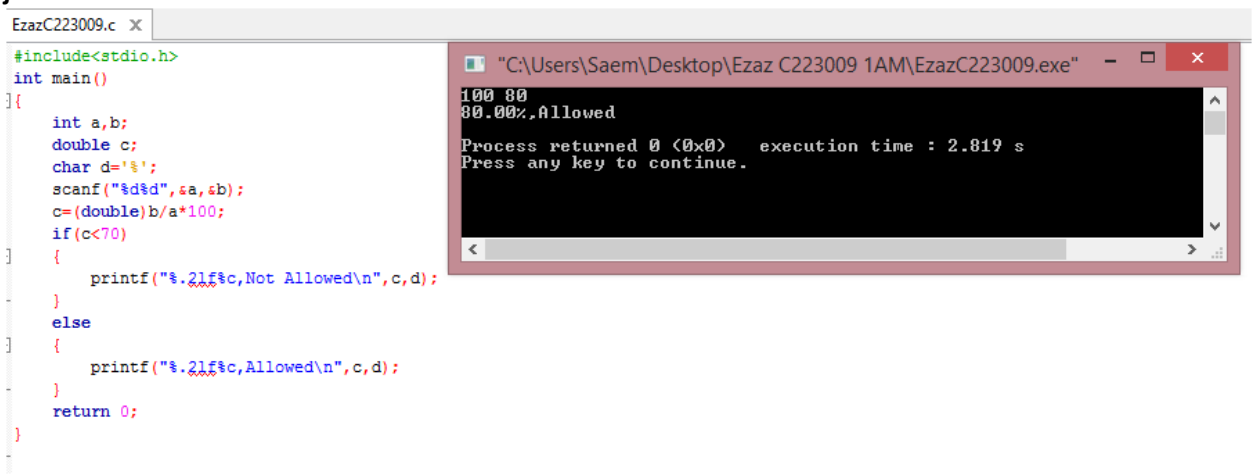
```
#include<stdio.h>
int main()
{
   int n;
   printf("Enter the value:");
   scanf("%d",&n);
   switch(n)
   {
   case 1:
      printf("Agrabad\n");
      break;
   case 2:
      printf("Chawkbazar\n");
      break;
   case 3:
      printf("Boddharhat\n");
      break;
   default:
      printf("Out of range of transport facility\n");
      break;
   }
   return 0;
}
```

In this code, if the user enters the value **1**, the program will print "Agrabad". If the user enters the value **2**, the program will print "Chawkbazar". If the user enters the value **3**, the program will print "Boddharhat".
If the user enters a value that is not in the range of 1 to 3, the program jumps to the **default** label and prints the message "Out of range of transport facility".
The **break** statements are used to exit the **switch** statement once a matching case has been found and its corresponding code has been executed. This prevents the program from executing any subsequent cases, even if the conditions are met.

No it's not true that "The default case is required in the switch statement".

3(d)-
Ans:

```
#include<stdio.h>
int main()
{
   int a,b;
   double c;
   char d='%';
   scanf("%d%d",&a,&b);
   c=(double)b/a*100;
   if(c<70)
   {
      printf("%.2lf%c,Not Allowed\n",c,d);
   }
   else
   {
      printf("%.2lf%c,Allowed\n",c,d);
   }
   return 0;
}
```



4(a)-
Ans:

(a) 10 0

(b) 10 20 30 40 50 60 70 80 90 100 110

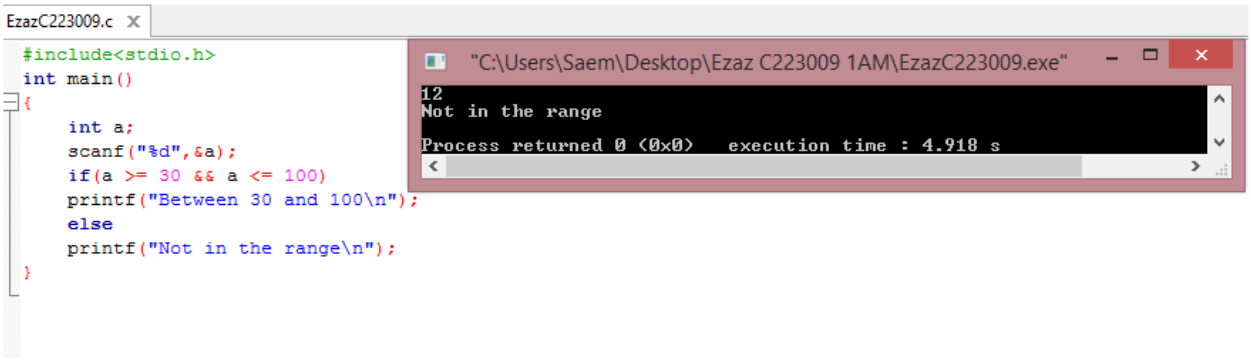
(c) 3 6

4(b)-
Ans:

The output of the program will always be "Between 30 and 100" for the different values of a.

The Correct code is given below-

```
#include<stdio.h>
int main()
{
   int a;
   scanf("%d",&a);
   if(a >= 30 && a <= 100)
   printf("Between 30 and 100\n");
   else
   printf("Not in the range\n");
}
```



4(c)-
Ans:

```
#include <stdio.h>
int main()
{
   int a,z;
   scanf("%d",&a);
   if (a%2==0)
   {
      z=1;
   }
   else
   {
      z=0;
   }
   printf("%d\n",z);
   return 0;
}
```

4(d)-

Ans:

```
#include <stdio.h>
int main()
{
   int N,n,count=0;
   printf("Enter a positive integer:  ");
   scanf("%d",&N);
   n=N;
   while (n!=0)
   {
      int r=n%2;
      if(r==1)
         count++;
      n/=2;
   }
   if (count%2==0)
      printf("%d is an Evil Number\n",N);
   else
      printf("%d is an Odious Number\n",N);
   return 0;
}
```

EzazC223009.c

```
#include <stdio.h>
int main()
{
    int N,n,count=0;
    printf("Enter a positive integer: ");
    scanf("%d",&N);
    n=N;
    while (n!=0)
    {
        int r=n%2;
        if(r==1)
            count++;
        n/=2;
    }
    if (count%2==0)
        printf("%d is an Evil Number\n",N);
    else
        printf("%d is an Odious Number\n",N);
    return 0;
}
```

"C:\Users\Saem\Desktop\Ezaz C223009 1AM\EzazC223009.exe"

```
Enter a positive integer: 4
4 is an Odious Number

Process returned 0 (0x0)   execution time : 1.098 s
Press any key to continue.
```

Thanks Everyone Assalamualikum