Tut. Sheet 2

Solutions @AyushR1

1. Write logical expressions that test whether a given character variable c is ➖

a. lower case letter

b. upper case letter

c. digit

d. white space (includes space, tab, new line)

Ans ➖

```
if (ch >= 'a' && ch <= 'z')
        printf("\n%c is an LowerCase character", ch);
    else if (ch >= 'A' && ch <= 'Z')
        printf("\n%c is an UpperCase character", ch);
    else if if(ch >= '0' && ch <= '9')
        printf("\n%c is a digit", ch);

    else if (ch == ‘ ‘ ||   /* ch is a blank space or */
         ch == ‘\t’ ||  /* a horizontal tab or */
         ch ==’\n’)  /* a new line or */
        printf("\n%c is a whitespace", ch);
```

2. Using precedence rules, evaluate the following expressions and determine the value of the variables (without running the code). Also, rewrite them using parenthesis to make the order explicit.

1. Assume(x=0xFF33,MASK=0xFF00).
    a. Expression:

```
c=x&MASK==0;
```

2. Assume(x=10,y=2,z=2;).
    a. Expression:

```
z=y=x++ + ++y∗2;
```

3. Assume(x=10,y=4,z=1;).
    a. Expression:

```
y >> = x & 0x2 && z;
```

Ans:-
(a) The operator precedence is '==' > '&' > '='.

Thus, the expression is equivalent to c= (x & (MASK==0)).
Therefore x=0xFF33,c=0.

(b) The operator precedence is '++' > '*' > '+'.

Thus, the expression is equivalent to z =(x++) + ((++y)∗2).
Therefore x=11,y=16,z=10+3∗2=16.

(c) The operator precedence is '&' > '&&' > '>>='.

Thus, the expression is equivalent to y>>= (x & 0x2) && z.
Therefore x=10,y=2,z=1.

3) Determine if the following statements have any errors. If so, highlight them and explain why.

- int 2ndvalue=10;
- Assume(x=0,y=0,alliszero=1).alliszero=(x=1)&&(y=0);

- Assume(x=10,y=3,z=0;).y=++x+y;z=z−−>x;

Ans:

1. The variable value should not start with a digit.
2. = operator should be replaced with ==.i,e alliszero=(x==1)&&(y==0).

Output :- 0

1. this is a confusing statement but it's correct. y=(++x)+y;z=(z−−)>x;

Output:- y=14 z=0

4. Both the for loop and the do-while loop can be transformed into a simple while loop. For the following example, write equivalent code using a while loop instead.

```
int factorial (int n) {

int i , ret = 1 ;

for ( i = 2 ; i <= n ; i++)

ret ∗= i ;

return ret ;

}
```

Ans:

```
int factorial (int n) {

int i=2 , ret = 1 ;

while (i<=n)
 {    ret*=i;
 i++;}

return ret ;
}
```

5. What will be the output of following programs?

a.

```
#include <stdio.h>

int main ()
{

  int a = 500, b, c;

  if (a >= 400)

    b = 300;

  c = 200;

  printf ("%d %d\n", b, c);

  return 0;

}
```

Ans:

300 200

b

```
#include <stdio.h>
int main ()
{
  int i = 65;
  char j = 'b';
  if (i == j)
    printf ("C is WOW\n");
  else
    printf ("C is a headache\n");
  return 0;
}
```

Ans

C is WOW

Given three points (x1, y1), (x2, y2) and (x3, y3), write a program to check if all the three points fall on one straight line.

```c
#include<stdio.h>
#include<math.h>
int main()
{
    int x1, y1, x2, y2, x3, y3;
    double ab, bc, ac, abc;

    printf("Enter the co-ordinates of first point (X1, Y1): ");
    scanf("%d %d", &x1, &y1);
    printf("Enter the co-ordinates of second point (X2, Y2): ");
    scanf("%d %d", &x2, &y2);
    printf("Enter the co-ordinates of third point (X3, Y3): ");
    scanf("%d %d", &x3, &y3);

    //suppose we have three points a, b, c
    //then all these points fall on one straight line if and only if
    //ab + bc = ac (distance should be same)

    ab = sqrt(pow(x2-x1,2)+pow(y2-y1,2));
    bc = sqrt(pow(x3-x2,2)+pow(y3-y2,2));
    ac = sqrt(pow(x3-x1,2)+pow(y3-y1,2));

    printf("ab: %f\t bc: %f\t ac: %f\n",ab, bc, ac);
    abc = ab+bc;
    if(abc==ac)
    {
        printf("ab + bc = ac\n");
        printf("All the three points fall on one straight line.");
    }
    else
        printf("All the three points are not present on one straight line.");

return 0;
}
```

7. If a = 10, b = 12, c = 0, find the values of the expressions in the following table:

a. a != 6 && b > 5

b. a == 9 || b < 3

c. ! (a < 10)

d. ! (a > 5 && c)

e. 5 && c != 8 || !c

Expressions	Value
a != 6 && b > 5	1
a == 9 || b < 3	0
!(a < 10)	1
!(a > 5 && c)	1
5 && c != 8 || !c	1