Assignment - 3

Ans-1. Ternary Operator: Teneary Operator perform the conditional operator and excute the the Code on the Basis of Binary condition.

Syntax → r= Condition-1 ? condition-2 : Condition-3

If condition1 is true then Condition-2 is executed otherwise Condition-3 will be execute.

// Example of Tenary Operator in C-programming

```
~~# include <stdio.h>~~
~~int main(){~~
~~result = num1>num2 ? num1 : num2~~
~~printf("%d", ~~num~~ result);~~
~~return 0;~~
~~}~~
```

```
# include <Stdio.h>
int main(){
    printf("Input Num1 and Num2");
    Scanf(" %d", &Num1,& Num2);
  int result = num1>num2 ? num1: num2;
  printf ("%d", result);
  return 0;
}
```

Special Operator :→ These Operators are used to perform logical operations.
Eg Mostly used Special operator in C programming language are:-
&, *, Size of() and ternary Operator.

(1) & = Is an Reference Operator which provide the address of the variable.
Eg &A will give Hexadecimal address of A.

(2) Size of() = Size of() Operator is used to find the Size of datatypes used.
Eg Size of (int) will give output as 4.

(3) * = star (astrick) is used to Pointers which stores the address of another Variable.
Eg &a *a will give the value present at this address.

(4) Ternary Operator = Ternary Operator are used to find the condition over the statement using the code on the basis of binary Conditions.

Ans-2:

```
# include <Stdio-h>
int main ( )
{
    int i,j,k;
    for(i=1; i<=3; i++){
        for (~~int~~j=1; j<=3 ; j++){
            for (int k=1; k<=3; k++){
                if(i!=j && j!=k&& j!=k){
                    printf ("%d %d %d", i,j,k);
                }
            }
        }
    }

    return 0;
}
```

Ans 3-

```
#include <stdio.h>
#include <string.h>
void namesorting()
{
    char name [5][30];
    Char temp [10];
    for (int i=0; i<5; i++){
        printf (" Enter %d name", i+1);
        Scanf ("%s", name [i]);
    }
    for (int j=0; j<4; j++){
        for (int k=j+1; k<5; k++){
            int r = strcmp (name [j], name [k]);
            if (r>0){
                strcpy (temp, name [j]);
                strcpy (name [j], name[k]);
                Strcpy (name [k], temp);
            }
        }
    }
    for (int j=0; j<5; j++){
        printf (~~"%d"~~ "Name %d is: %s", j+1, name[j] );
    }

    printf ("\n");
    Char key [10];
    printf ("Input as key to search names");
    Scanf ("%s", &key);
    for (int k=0; k<5; k++){
        int m= strcmp (name [k], key);
        if (m==0){
            printf (" Name found ! Name is %s ", name[i]),
        }   break;
        else if (i==4){
            printf (" Name Not found");
        }
    }
};

int main ()
{
    namesorting ();
    return 0;
}
```

Ans-4-

```
#include <Stdio.h>
Struct Students
{
    char * name;
    int enroll;
    double SapId;
};

int main()
{
    Students a1[5];
    for(int i=0;i<5; i++){
        printf(" Enter %d details", i+1);
        Scanf ("%S %d %d", &a1[i].enroll, &a1[i].SapId;
                                     &a1[i].name);

    }

    for (int k=0; k<5; k++){
        printf("Name:%S, Enrollement No: %d,
                    SapId: %d \n", a1[i]. name, a1[i]. enroll;
                                     a[i]. sapid);
    }
    return0;
}
```

```
        return (n%2) +10 *  con.
```

Ans.5-

```
# include <stdio.h>
int binary (int num)
{
        if (n==0){
              return 0;
        }
        else{
            return (n%2) + 10 * binary (n/2);
        }
}

int main()
{
      int n;
      printf("Enter Number:");
      scanf ("%d", &n);
      printf ("Binary Equivalent : %d", binary (n));
}
```

Ans-6 -

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    Char str [1000];
    FILE *fptr;
    fptr. fopen("IBM.txt", "w");
    if (fptr== null)
    {
        printf("error");
        exit(1);
    }
    printf ("Enter Data into IBM:\n");
    fgets (Sentence, sizeof (~~sen~~ str), stdin);
    fprintf (fptr, "%s", str);
    fclose(fptr);

    Char * filename = "IBM.txt";
    FILE * fp = fopen (filename, "r");
    if (fp == null)
    {
        printf ("Error")
        return1;
    }
    Char ch;
    While ((ch = fgetc (f)) != EOF
            put char (ch);
    fclose (fp);
    return 0;
}
```