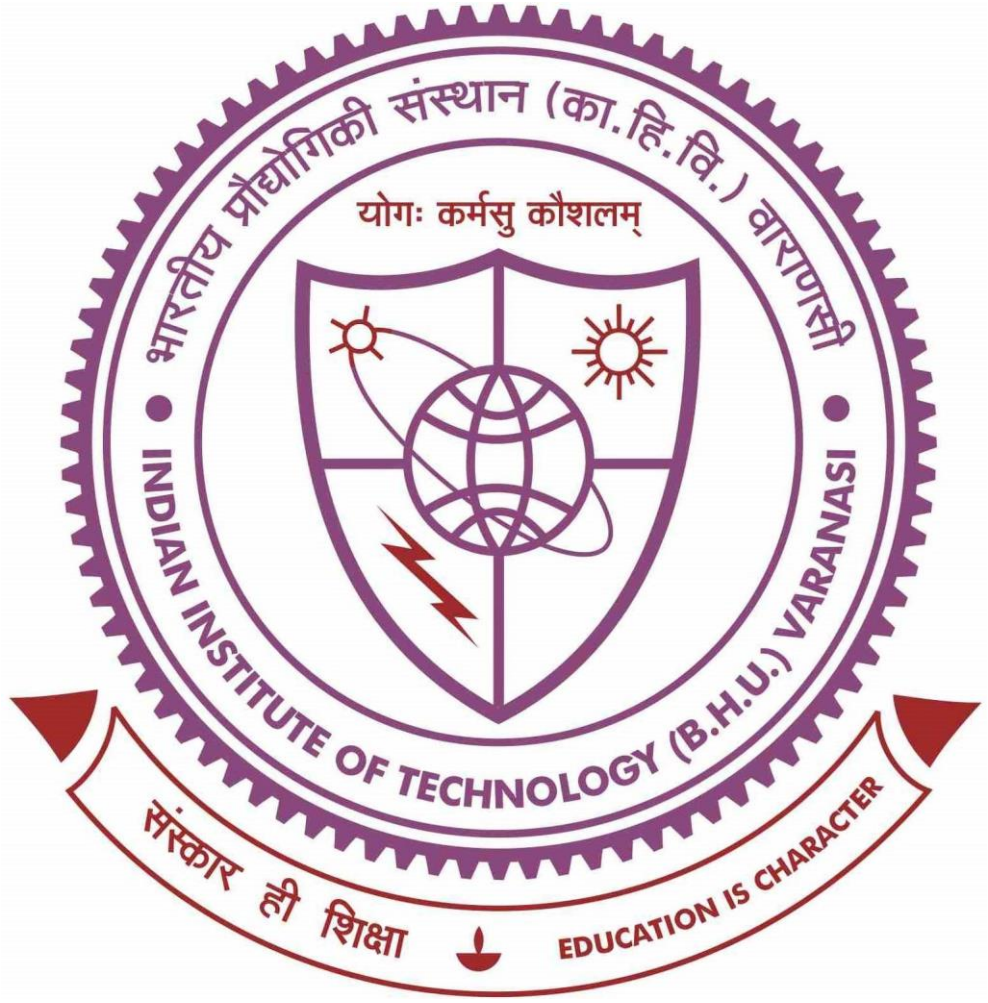

CSO-101 ASSIGNMENT 9

Name- Partik Singh Bumrah

Roll No-21075064

Dept-CSE

Section-BA 1

1. **Matching the following:-**

Solution:-

1-f	6-i
2-j	7-d
3-m	8-k
4-c	9-h
5-a	10-e

1. **Write a program to insert a substring into another string by using function and pointers.**

Solution:-

```c
#include <stdio.h>
#include <string.h>
void join(char *str, char str1[], int x)
{

    int l1 = strlen(str), l2 = strlen(str1);
    char tmp[l1];
    for (int i = 0; i < l1; i++)
    {
        tmp[i] = str[i];
    }

    for (int k = x + l2, p = x; k < l1 + l2, p < l1; k++, p++)
    {
        str[k] = str[p];
    }
    for (int i = x, j = 0; i < x + l2, j < l2; i++, j++)
    {
        str[i] = str1[j];
    }
    int t = 0, o = l2 + x;
    char j;
    for (int i = x; i < l1 + l2; i++)
    {
        j = tmp[i];
        if (t < l2)
        {
            str[i] = str1[t];
```

```
            t += 1;
        }
        str[o] = j;
        o = o + 1;
    }
}
int main()
{
    char str1[200], str2[100];
    int x;
    printf("Enter the first string:");
    gets(str1);
    printf("Enter the substring that you want to insert:");
    gets(str2);
    printf("Enter the index from where you want to insert the
substring:");
    scanf("%d", &x);
    join(str1, str2, x);
    printf("%s", str1);
    return 0;
}
```

2. **Write a program using pointers to read in an array of integers and print its elements in reverse order.**

Solution:-

```
#include <stdio.h>
int main()
{
    int n;
    printf("Enter the size of the array :");
    scanf("%d",&n);
    int arr[n];
    int *ptr[n];
     for (int i = 0; i < n; i++)
     {
         scanf("%d",&arr[i]);
         ptr[i]=&arr[i];
     }
     for (int i = n-1; i >=0; i--)
     {
         printf("%d ",*ptr[i]);
     }

    return 0;
```

```
}
```

3. **Using pointers, write a function that receives a character string and a character as argument anddeletes all occurrences of this character in this string. The returned string should not have any holes.**

Solution:-

```c
#include <stdio.h>
#include <string.h>
void del(char* str, char a)
{

    int j;
    for (int i = j = 0; i <strlen(str); i++)
    {
        if (str[i] != a)
        {
            str[j] = str[i];
            j++;
        }
    }

    str[j] = '\0';
}

int main()
{
    char str[100],ch;
    printf("Enter the string:");
    gets(str);
    printf("Enter the value of the character:");
    scanf("%c",&ch);
    del(str, ch);
    printf("%s",str);
    return 0;
}
```

4. **Write a C program that uses the pointer increment operations to demonstrate the scale factor.**

Solution:-

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

```
     int a,*ptr;
     ptr=&a;
     printf("The size of the int data type is %d\n",sizeof(int));
     printf("The address of the variable before the increment is
%p\n",ptr);
     ++ptr;
     printf("The address of the variable after the increment is
%p\n",ptr);
    return 0;
 }
```

5. **Write a C program that displays the addresses and values pointed by an array of integer pointers.**

Solution:-

```
#include<stdio.h>
int main()
{
    int n;
    printf("Enter the number of elements of a array:");
    scanf("%d",&n);
    int arr[n];
    int *ptr[n];
    printf("Enter the elements of the array:\n");
    for (int i = 0; i < n; i++)
    {
        scanf("%d",&arr[i]);
        ptr[i]=&arr[i];
    }
    printf("The values of the array are                    The values of
addresses are\n");
    for (int i = 0; i < n; i++)
    {
        printf("%d                                                %p\n",*
ptr[i],ptr[i]);
    }
    return 0;
}
```

6. **Write a C program that demonstrates the difference between pass by value and pass by reference.**

Solution:-

```c
#include<stdio.h>
int sum_by_value(int x,int y)
{
    int sum=x+y;
    return sum;
}
int sum_by_reference(int *p,int *q)
{
    int sum = *p+ *q;
    return sum;
}
int main()
{
     int x,y;
     printf("Enter the value of a 2 numbers to find their sum:");
     scanf("%d %d",&x,&y);
     printf("The value of the sum by call by value
%d\n",sum_by_value(x,y));
     printf("The value of the sum by reference by value
%d\n",sum_by_reference(&x,&y));
    return 0;
}
```

7. Write a C program that checks whether two strings are equal by using pointers.

Solution:-

```c
#include<stdio.h>
#include<string.h>
int main()
{
    char str1[100];char str2[100];int flag=1;
    printf("Enter the string 1:");gets(str1);
    printf("Enter the string 2:");gets(str2);
    if (strlen(str1)!=strlen(str2))flag=0;
    else{
        for (int i = 0; i < strlen(str1); i++)
        {
            char *p1= &str1[i];
            char *p2= &str2[i];
            if (*p1!=*p2)
            {
                flag=0;
            }

        }
```

```
    }
    if (flag==0)printf("The strings are not equal");
    else printf("The strings are equal");
     return 0;
}
```

8. Write a C program that demonstrates the difference between array of pointers and pointer to an array.

Solution:-

```
#include<stdio.h>
int main()
 {
     int n;
     printf("Enter the length of the array :");scanf("%d",&n);
     int ar[n],(*ptr1)[n],*ptr2[n];printf("Input the elements of the
integer array\n");
     for (int i = 0; i < n; i++)
     {
        scanf("%d",&ar[i]);
        ptr2[i]=&ar[i];
     }
     ptr1=&ar;
     printf("Printing the elements of the array by using pointers to
array\n");
     for (int i = 0; i < n; i++) printf("%d\n",(*ptr1)[i]);
     printf("Printing the elements of the array by using array of
pointers \n");
     for (int i = 0; i < n; i++)printf("%d\n",*ptr2[i]);
       return 0;
 }
```