**May 12 Lab questions**

#1. WAP to compute the sum of all elements in an array using pointer.

Code:

#include <stdio.h>

int main()

{

    int i\_285,n\_285, \*p,sum;

    int arr[20];

    printf("Provide the size of your array\n");

    scanf("%d",&n\_285);

    printf("Give the value of your array\n");

    for(i\_285=0;i\_285<n\_285;i\_285++)

    {

        scanf("%d",&arr[i\_285]);

    }

    p=arr;

    for(i\_285=0;i\_285<n\_285;i\_285++)

    {

        sum=sum+\*p;

        \*p++;

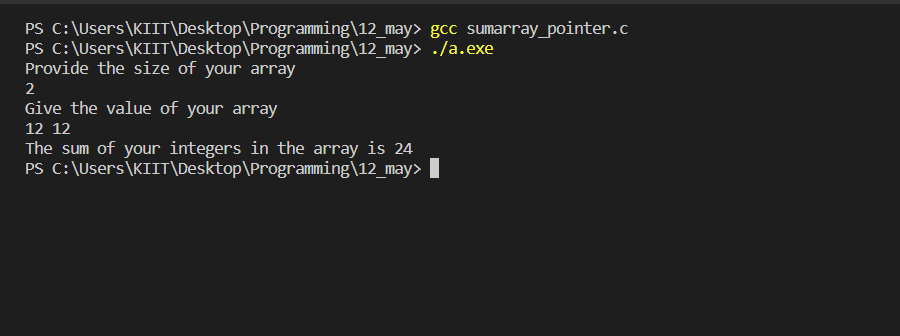
    }

    printf("The sum of your integers in the array is %d\n",sum);

    return 0;

}

Output:



#2. WAP to display values in reverse order from an integer array using pointer.

Code:

#include <stdio.h>

int main()

{

    int num\_285,arr[30],\*ptr\_285,n,i,j;

    printf("Please provide your integer\n");

    scanf("%d",&num\_285);

    while(num\_285!=0)

    {

        arr[i]=num\_285%10;

        num\_285=num\_285/10;

        i++;

    }

    ptr\_285=arr;

    for(j=i;j>0;j--)

    {

        printf("%d ",\*ptr\_285);

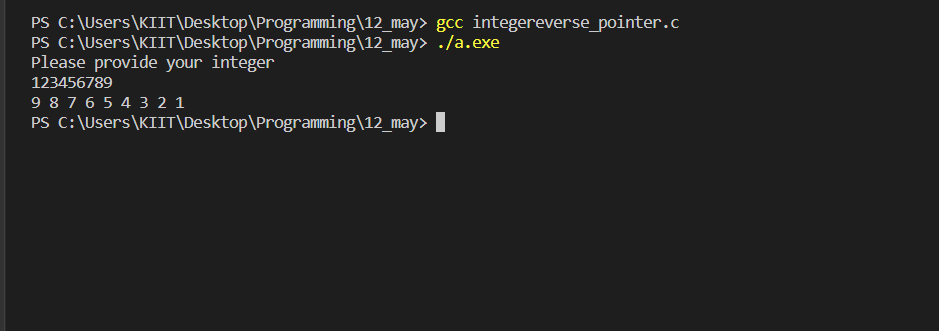
        \*ptr\_285++;

    }

    return 0;

}

Output:



#3. WAP to swap three numbers in cyclic order using Call by Reference. In

other words, WAP that takes three variable (a, b, c) in as separate parameters

and rotates the values stored so that value a goes to be, b, to c and c to a.

Code:

#include <stdio.h>

int main()

{

    int a,b,c, \*ptra\_285,\*ptrb\_285,\*ptrc,temp;

    printf("please provide the value of your 1st,2nd and 3rd integers respectively;\n");

    scanf("%d%d%d",&a,&b,&c);

    ptra\_285=&a; ptrb\_285=&b; ptrc=&c;

    printf("The value of your 1st integer before swap is=%d\n2nd integer=%d\n3rd integer=%d\n",\*ptra\_285,\*ptrb\_285,\*ptrc);

    temp= \*ptra\_285;

    \*ptra\_285=\*ptrb\_285;

    \*ptrb\_285=\*ptrc;

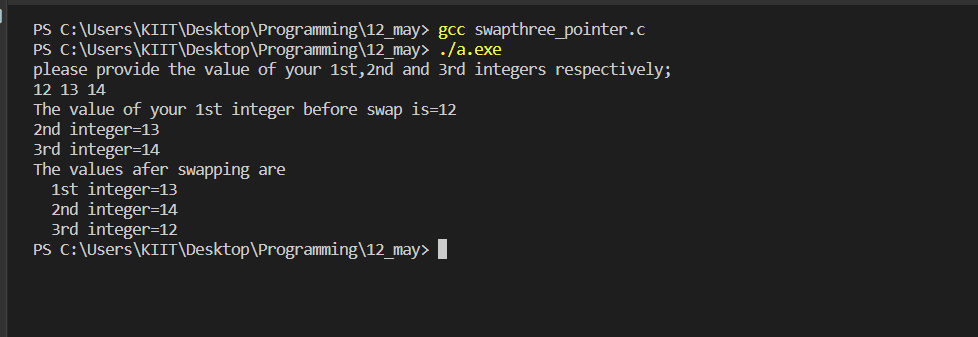
    \*ptrc=temp;

    printf("The values afer swapping are\n  1st integer=%d\n  2nd integer=%d\n  3rd integer=%d\n",\*ptra\_285,\*ptrb\_285,\*ptrc);

    return 0;

}

Output:



#4.WAP to sort an array using Pointer.

Code:

#include <stdio.h>

int main()

{

    int i\_285,j\_285,arr\_285[30],n,\*ptr,temp\_285;

    printf("please provide the size of your list of numbers\n");

    scanf("%d",&n);

    printf("Provide the elements of your list\n");

    for(i\_285=0;i\_285<n;i\_285++)

    {

        scanf("%d",&arr\_285[i\_285]);

    }

    ptr=&arr\_285[0];

    for(i\_285=0;i\_285<n-1;i\_285++)

    {

        for(j\_285=i\_285+1;j\_285<n;j\_285++)

        {

            if(\*(ptr+i\_285)< \*(ptr+j\_285))

            {

                temp\_285= \*(ptr+i\_285);

                \*(ptr+i\_285)= \*(ptr+j\_285);

                \*(ptr+j\_285)=temp\_285;

            }

        }

    }

    printf("\nThe elements in the array after sorting : \n\n");

    i\_285=0;

    while(i\_285<n)

    {

        printf("%d ",\*(ptr+i\_285));

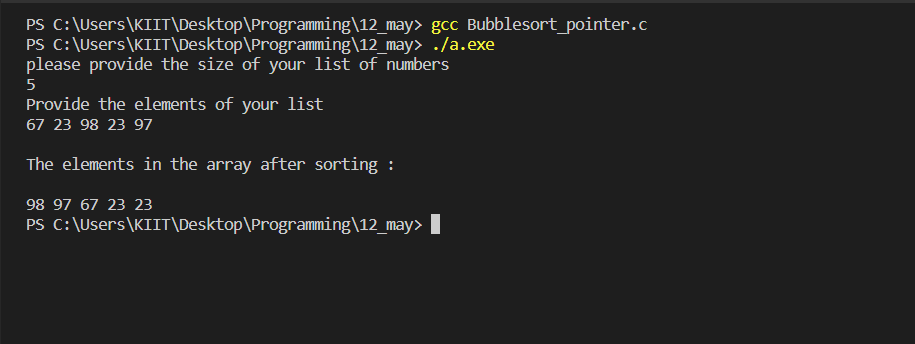
        i\_285++;

    }

    return 0;

}

Output:



#5.WAP to count vowels and consonants in a string using pointer.

Code:

#include <stdio.h>

#include <string.h>

int main()

{

    char arr\_285[20], \*ptr\_285;

    int i,n,cCount\_285,vCount\_285;

    printf("Provide your string\n");

    fgets(arr\_285,20,stdin);

    ptr\_285=&arr\_285[0];

    n=strlen(arr\_285);

    while(\*ptr\_285!='\0')

    {

        if(\*ptr\_285=='U'|| \*ptr\_285=='O'|| \*ptr\_285=='I'|| \*ptr\_285=='E'|| \*ptr\_285=='A'|| \*ptr\_285=='u'|| \*ptr\_285=='o'|| \*ptr\_285=='i'|| \*ptr\_285=='e'|| \*ptr\_285=='a')

            vCount\_285++;

        ptr\_285++;

    }

    cCount\_285=n-vCount\_285-1;

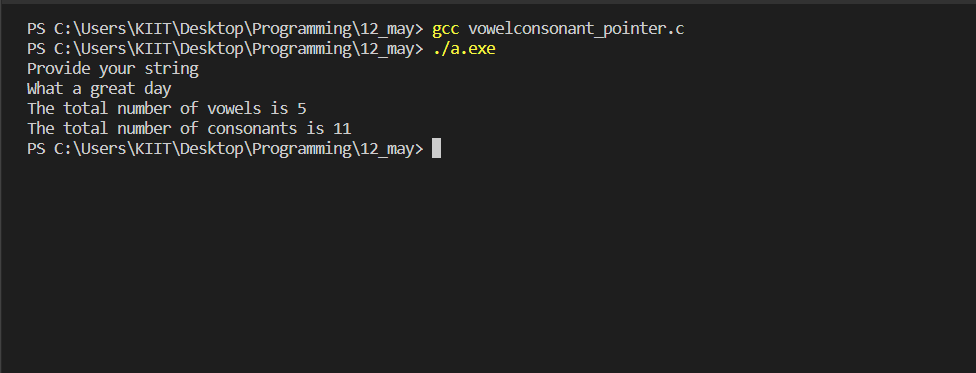
    printf("The total number of vowels is %d\n",vCount\_285);

    printf("The total number of consonants is %d\n",cCount\_285);

    return 0;

}

Output:



#6.WAP to print a string in reverse using a pointer.

Code:

#include <stdio.h>

#include <string.h>

int main()

{

    char arr\_285[30], \*ptr\_285;

    int n,i;

    printf("Provide a short string\n");

    fgets(arr\_285,30,stdin);

    n=strlen(arr\_285);

    ptr\_285=&arr\_285[n];

    for(i=0;i<=n;i++)

    {

        printf("%c ",\*ptr\_285);

        ptr\_285--;

    }

    return 0;

}

Output:

