7. Programs Using Pointers

7.1.Area and perimeter of circle using pointers

#include <stdio.h>

float pi = 3.14;

float area\_n\_peri(float a,float \*ar,float \*per){

    \*ar = pi \* a \* a;

    \*per = 2 \* pi \* a;

}

int main(){

    float r,area,peri;

    printf("enter the radius: ");

    scanf("%f",&r);

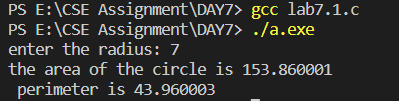
    area\_n\_peri(r,&area,&peri);

    printf("the area of the circle is %f\n perimeter is %f",area,peri);

    return 0;

}

Compiler –



7.2.To check whether a number is prime using function pointers

#include <stdio.h>

int main(){

    int a;

    printf ("enter a number: ");

    scanf("%d",&a);

    int \*ptr = &a;

    for(int i = a-1; i>1 ; i--)

    {

        if(\*ptr%i == 0)

            {

                printf("it is not a prime number");

                return 0;

            }

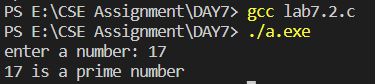
    }

    printf("%d is a prime number",\*ptr);

    return 0;

}

Compiler –



7.3.Duplication Removal using pointer

#include <stdio.h>

#include <string.h>

int main(){

    int n;

    printf("enter the size of the array: ");

    scanf("%d",&n);

    int ab[n];

    int bc[n];

    int \*ptr = ab;

    int \*ptr2 = bc;

    for (int i = 0; i < n; i++)

    {

        printf("enter the element %d : ",(i + 1));

        scanf("%d",&ab[i]);

    }

    bc[0] = \*ptr;

    int k = 1;

    for (int i = 1; i<n; i++)

    {

        for (int j = 0; j < i; j++)

        {

            if(\*(ptr+i) == \*(ptr2+j))

                {j = i;}

            if (j == (i-1))

                {

                    bc[k] = \*(ptr+i);

                    k++;

                }

        }

    }

    for (int i = 0; i < k; i++)

    {

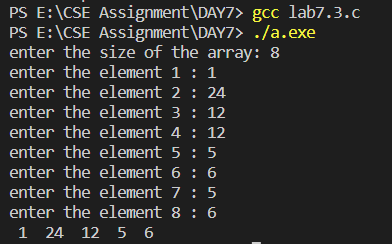
        printf(" %d ",\*(ptr2+i));

    }

    return 0;

}

Compiler –



7.4. Sorting integer arrays using pointers

#include <stdio.h>

int main(){

    int n;

    printf("enter the size of the array: ");

    scanf("%d",&n);

    int abc[n];

    int \*ptr = abc;

    for (int i = 0; i < n; i++)

    {

        printf("enter the element %d : ",(i + 1));

        scanf("%d",(ptr+i));

    }

    int temp;

    for(int i = 0; i<n; i++)

    {

        for(int j = i; j<n; j++)

        {

            if (\*(ptr+i)>\*(ptr+j))

                {

                    temp = \*(ptr+i);

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

                    \*(ptr+j) = temp;

                }

        }

    }

    for (int i = 0; i<n; i++)

    {

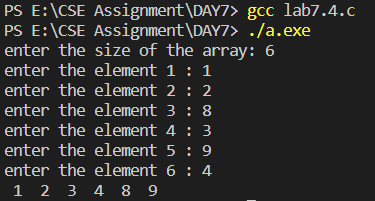
        printf(" %d ",\*(ptr+i));

    }

    return 0;

}

Compiler –



7.5.Sum of array using pointers

#include <stdio.h>

int main(){

    int n;

    printf("enter the size of the array: ");

    scanf("%d",&n);

    int abc[n];

    int \*ptr = abc;

    for (int i = 0; i < n; i++)

    {

        printf("enter the element %d : ",(i + 1));

        scanf("%d",(ptr+i));

    }

    int sum = 0;

    for(int i = 0; i<n; i++)

    {

        sum += \*(ptr+i);

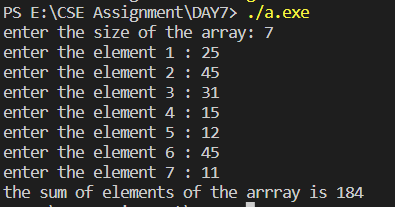
    }

    printf("the sum of elements of the arrray is %d",sum);

    return 0;

}

Compiler –



7.6.Count number of space, words, digits, numbers, vowels using pointers

#include <stdio.h>

#include <ctype.h>

void count(char \*str, int \*spaces, int \*words, int \*digits, int \*vowels) {

    int i = 0, newWord = 1;

    while (str[i] != '\0') {

        if (isspace(str[i])) {

            (\*spaces)++;

            newWord = 1;

        } else if (newWord) {

            (\*words)++;

            newWord = 0;

        }

        if (isdigit(str[i])) {

            (\*digits)++;

        }

        if (str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u' ||

            str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U') {

            (\*vowels)++;

        }

        i++;

    }

}

int main() {

    char str[100];

    int spaces = 0, words = 0, digits = 0, vowels = 0;

    printf("Enter a string: ");

    gets(str);

    count(str, &spaces, &words, &digits, &vowels);

    printf("Spaces: %d\n", spaces);

    printf("Words: %d\n", words);

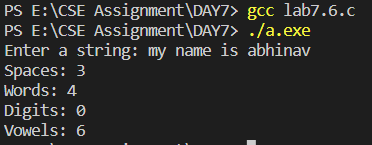
    printf("Digits: %d\n", digits);

    printf("Vowels: %d\n", vowels);

    return 0;

}

Compiler -



7.7. Length of a string using pointer

#include <stdio.h>

#include <string.h>

int main(){

    char abc[50];

    printf("enter a string: ");

    gets(abc);

    char \*ptr = abc;

    int count = 0;

    for (int i = 0; \*(ptr+i) != '\0'; i++)

    {

        count++;

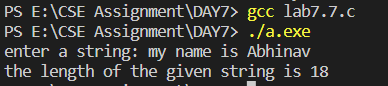
    }

    printf("the length of the given string is %d",count);

    return 0;

}

Compiler –



7.8.Reverse the String Using Pointers

#include <stdio.h>

int main(){

    int n;

    printf("enter the size of the array: ");

    scanf("%d",&n);

    int a = n;

    int abc[n];

    int \*ptr = abc;

    for (int i = 0; i < n; i++)

    {

        printf("enter the element %d : ",(i + 1));

        scanf("%d",(ptr+i));

    }

    int temp;

    for (int i = 0; i<= n/2; i++,n--)

    {

        temp = \*(ptr+i);

        \*(ptr+i) = \*(ptr+(n-1));

        \*(ptr+(n-1)) = temp;

    }

    for(int i = 0; i<a; i++)

    {

        printf(" %d ",\*(ptr+i));

    }

    return 0;

}

Compiler –

