**26/03/2024**

**Program -1**

#include <stdio.h>

int main()

{

int arr[]={20,30,40,50};

int \*ptr=arr;

int size = sizeof(arr) / sizeof(arr[0]);

for(int i=0;i<size;i++)

{

printf("%d\n",\*(ptr+i));

}

return 0;

}

**Program 2**

#include <stdio.h>

int main()

{

char str[] = "Hello, Pointers!";

char \*ptr =str;

char size = sizeof(str) / sizeof(str[0]);

for(int i=0;i<size;i++)

{

printf("%c\n",\*(ptr+i));

}

return 0;

}

**Program 3**

void square(int\* num)

{

\*num=(\*num) \* (\*num);

}

int main()

{

int x = 5;

square(&x);

printf("Modified value of x:%d",x);

return 0;

}

**Program 4**

#include <stdio.h>

void swap(int \*a, int \*b)

{

int temp\_var=\*a;

\*a=\*b;

\*b=temp\_var;

}

int main()

{

int x = 10, y = 20;

swap(&x,&y);

printf("After swapping values of x and y: %d %d",x,y);

return 0;

}

**Program 5:**

**scenario:**

**You are tasked with creating a simple program that sorts an array of integers using different sorting algorithms. However, instead of implementing each sorting algorithm separately, you are required to utilize function pointers to switch between them dynamically.**

**Write a C program that implements three sorting algorithms: bubble sort, selection sort, and insertion sort. Define each sorting function to take an integer array and its size as arguments and sort the array in ascending order. Additionally, declare a function pointer type named SortFunction that points to functions with the same signature.**

**Your program should prompt the user to enter the number of elements in the array and the elements themselves. Then, it should display a menu allowing the user to choose the sorting algorithm they want to use. Based on the user's choice, use the appropriate function pointer to sort the array and display the sorted result.**

**Output:**

#include <stdio.h>

void Bubblesort( int arr[], int num\_of\_elements)

{

int temp;

for(int i=0;i<num\_of\_elements;i++)

{

for(int j=i+1;j<num\_of\_elements;j++)

{

if(arr[i]>arr[j])

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

}

void Insertionsort(int arr[], int num\_of\_elements)

{

int i,j,temp;

for (i = 1; i < num\_of\_elements; i++)

{

temp = arr[i];

j = i - 1;

while(j>=0 && temp <= arr[j])

{

arr[j+1] = arr[j];

j = j-1;

}

arr[j+1] = temp;

}

}

void Selectionsort(int arr[], int num\_of\_elements)

{

int i, j,min;

for (i = 0; i < num\_of\_elements-1; i++)

{

min = i;

for (j = i+1; j < num\_of\_elements; j++)

if (arr[j] < arr[min])

min = j;

int temp = arr[min];

arr[min] = arr[i];

arr[i] = temp;

}

}

void getInputArray(int arr[],int num\_of\_elements)

{

for(int i=0;i<num\_of\_elements;i++)

{

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

}

}

int main()

{

int choice,num\_of\_elements;

printf("Enter the number of elements : ");

scanf("%d",&num\_of\_elements);

int arr[num\_of\_elements];

getInputArray(arr,num\_of\_elements);

printf("choice 1 - bubble sort\n");

printf("choice 2 - Insertion sort\n");

printf("choice 3 - Selection sort\n");

printf("enter the choice :");

scanf("%d",&choice);

void (\*SortFunction)(int \*, int);

switch(choice)

{

case 1 :

SortFunction = &Bubblesort;

break;

case 2 :

SortFunction = &Insertionsort;

break;

case 3 :

SortFunction = &Selectionsort;

break;

default:

printf("Invalid choice..");

}

SortFunction(arr, num\_of\_elements);

printf("Sorted Array elements :");

for(int i=0;i<num\_of\_elements;i++)

{

printf("%d ",arr[i]);

}

}

**Program 6:**

**The calculateSumAndProduct function takes two numbers (num1 and num2) and two pointers (sum and product) as parameters.**

**Inside the function, it calculates the sum and product of the two numbers and stores the results at the memory locations pointed to by sum and product.**

**In the main function, we declare variables sum and product.**

**We then call the calculateSumAndProduct function, passing the values of num1 and num2, as well as the addresses of sum and product.**

**After the function call, the values of sum and product are updated, and we can print the results.**

**Output:**

#include<stdio.h>

calculateSumAndProduct(int num1,int num2,int \*sum,int \*product)

{

\*sum=num1+num2;

\*product=num1\*num2;

}

int main()

{

int num1,num2,sum,product;

printf("Enter two numbers: ");

scanf("%d %d",&num1,&num2);

calculateSumAndProduct(num1,num2,&sum,&product);

printf("sum=%d,product=%d",sum,product);

}