**1. Write a program in C to add two numbers using pointers.**

#include <stdio.h>

main()

{

int fno, sno, \*ptr, \*qtr, sum;

printf("\n\n Pointer : Add two numbers :\n");

printf("--------------------------------\n");

printf(" Input the first number : ");

scanf("%d", &fno);

printf(" Input the second number : ");

scanf("%d", &sno);

ptr = &fno;

qtr = &sno;

sum = \*ptr + \*qtr;

printf(" The sum of the entered numbers is : %d\n\n",sum);

}

Output:

Pointer : Add two numbers :

--------------------------------

Input the first number : 5

Input the second number : 6

The sum of the entered numbers is : 11

2. Write a program in C to find the maximum number between two numbers using a pointer.

#include <stdio.h>

#include <stdlib.h>

main()

{

int fno,sno,\*ptr1=&fno,\*ptr2=&sno;

printf("\n\n Pointer : Find the maximum number between two numbers :\n");

printf("------------------------------------------------------------\n");

printf(" Input the first number : ");

scanf("%d", ptr1);

printf(" Input the second number : ");

scanf("%d", ptr2);

if(\*ptr1>\*ptr2)

printf("\n\n %d is the maximum number.\n\n",\*ptr1);

else

printf("\n\n %d is the maximum number.\n\n",\*ptr2);

}

Output:

Pointer : Find the maximum number between two numbers :

------------------------------------------------------------

Input the first number : 5

Input the second number : 6

6 is the maximum number.

**3. Write a program in C to find the largest element using Dynamic Memory Allocation.**

#include <stdio.h>

#include <stdlib.h>

main()

{

int i,n;

float \*element;

printf("\n Pointer : Find the largest element using Dynamic Memory Allocation :\n");

printf("----------------------------------------------------------------------\n");

printf(" Input total number of elements(1 to 100): ");

scanf("%d",&n);

element=(float\*)calloc(n,sizeof(float));

if(element==NULL)

{

printf(" No memory is allocated.");

exit(0);

}

printf("\n");

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

{

printf(" Number %d: ",i+1);

scanf("%f",element+i);

}

for(i=1;i<n;++i)

{

if(\*element<\*(element+i))

\*element=\*(element+i);

}

printf(" The Largest element is : %.2f \n\n",\*element);

}

Output:

Pointer : Find the largest element using Dynamic Memory Allocation :

-------------------------------------------------------------------------

Input total number of elements(1 to 100): 5

Number 1: 5

Number 2: 7

Number 3: 2

Number 4: 9

Number 5: 8

The Largest element is : 9.00

**4. Write a program in C to Calculate the length of the string using a pointer.**

#include <stdio.h>

int calculateLength(char\*);

main()

{

char str1[25];

int l;

printf("\n\n Pointer : Calculate the length of the string :\n");

printf("---------------------------------------------------\n");

printf(" Input a string : ");

fgets(str1, sizeof str1, stdin);

l = calculateLength(str1);

printf(" The length of the given string %s is : %d ", str1, l-1);

printf("\n\n");

}

Output:

Pointer : Calculate the length of the string :

---------------------------------------------------

Input a string : BITresource

The length of the given string BITresource

is : 11

**5. Write a program in C to swap elements using call by reference.**

#include <stdio.h>

void swapNumbers(int \*x,int \*y,int \*z);

main()

{

int e1,e2,e3;

printf("\n\n Pointer : Swap elements using call by reference :\n");

printf("------------------------------------------------------\n");

printf(" Input the value of 1st element : ");

scanf("%d",&e1);

printf(" Input the value of 2nd element : ");

scanf("%d",&e2);

printf(" Input the value of 3rd element : ");

scanf("%d",&e3);

printf("\n The value before swapping are :\n");

printf(" element 1 = %d\n element 2 = %d\n element 3 = %d\n",e1,e2,e3);

swapNumbers(&e1,&e2,&e3);

printf("\n The value after swapping are :\n");

printf(" element 1 = %d\n element 2 = %d\n element 3 = %d\n\n",e1,e2,e3);

}

void swapNumbers(int \*x,int \*y,int \*z)

{

int tmp;

tmp=\*y;

\*y=\*x;

\*x=\*z;

\*z=tmp;

}

Pointer : Swap elements using call by reference :

------------------------------------------------------

Input the value of 1st element : 5

Input the value of 2nd element : 6

Input the value of 3rd element : 7

The value before swapping are :

element 1 = 5

element 2 = 6

element 3 = 7

The value after swapping are :

element 1 = 7

element 2 = 5

element 3 = 6

**6. Write a program in C to find the factorial of a given number using pointers.**

#include <stdio.h>

void findFact(int,int\*);

main()

{

int fact;

int num1;

printf("\n\n Pointer : Find the factorial of a given number :\n");

printf("------------------------------------------------------\n");

printf(" Input a number : ");

scanf("%d",&num1);

findFact(num1,&fact);

printf(" The Factorial of %d is : %d \n\n",num1,fact);

}

void findFact(int n,int \*f)

{

int i;

\*f =1;

for(i=1;i<=n;i++)

\*f=\*f\*i;

}

Output:

Pointer : Find the factorial of a given number :

------------------------------------------------------

Input a number : 5

The Factorial of 5 is : 120

**7. Write a program in C to count the number of vowels and consonants in a string using a pointer.**

#include <stdio.h>

main()

{

char str1[50];

char \*pt;

int ctrV,ctrC;

printf("\n\n Pointer : Count the number of vowels and consonants :\n");

printf("----------------------------------------------------------\n");

printf(" Input a string: ");

gets(str1);

//assign address of str1 to pt

pt=str1;

ctrV=ctrC=0;

while(\*pt!='\0')

{

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

ctrV++;

else

ctrC++;

pt++; //pointer is increasing for searching the next character

}

printf(" Number of vowels : %d\n Number of consonants : %d\n",ctrV,ctrC-1);

}

Output:

Pointer : Count the number of vowels and consonants :

----------------------------------------------------------

Input a string: string

Number of vowels : 1

Number of consonants : 5

**8. Write a program in C to compute the sum of all elements in an array using pointers.**

#include <stdio.h>

main()

{

int arr1[10];

int i,n, sum = 0;

int \*pt;

printf("\n\n Pointer : Sum of all elements in an array :\n");

printf("------------------------------------------------\n");

printf(" Input the number of elements to store in the array (max 10) : ");

scanf("%d",&n);

printf(" Input %d number of elements in the array : \n",n);

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

{

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

scanf("%d",&arr1[i]);

}

pt = arr1;

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

{

sum = sum + \*pt;

pt++;

}

printf(" The sum of array is : %d\n\n", sum);

}

**Output:**

Pointer : Sum of all elements in an array :

------------------------------------------------

Input the number of elements to store in the array (max 10) : 5

Input 5 number of elements in the array :

element - 1 : 2

element - 2 : 3

element - 3 : 4

element - 4 : 5

element - 5 : 6

The sum of array is : 20

Write a program in C to print a string in reverse using a pointer.

#include <stdio.h>

main()

{

char str1[50];

char revstr[50];

char \*stptr = str1;

char \*rvptr = revstr;

int i=-1;

printf("\n\n Pointer : Print a string in reverse order :\n");

printf("------------------------------------------------\n");

printf(" Input a string : ");

scanf("%s",str1);

while(\*stptr)

{

stptr++;

i++;

}

while(i>=0)

{

stptr--;

\*rvptr = \*stptr;

rvptr++;

--i;

}

\*rvptr='\0';

printf(" Reverse of the string is : %s\n\n",revstr);

}

Output:

Pointer : Print a string in reverse order :

------------------------------------------------

Input a string : BITresource

Reverse of the string is : ecruoserTIB