# CSE A Pointers Examples

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

Test Data :

Input the first number : 5

Input the second number : 6

*Expected Output* :

```
The sum of the entered numbers is : 11
```

```c
#include <stdio.h>
int 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);

   return 0;
}
```

Sample 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 add numbers using **call by reference**.

Test Data :

Input the first number : 5

Input the second number : 6

*Expected Output* :

```
 The sum of 5 and 6  is 11
```

```c
#include <stdio.h>
long addTwoNumbers(long *, long *);

int main()
{
   long fno, sno, *ptr, *qtr, sum;

   printf("\n\n Pointer : Add two numbers using call by reference:\n");
   printf("-------------------------------------------------------\n");

   printf(" Input the first number : ");
   scanf("%ld", &fno);
   printf(" Input the second  number : ");
   scanf("%ld", &sno);
   sum = addTwoNumbers(&fno, &sno);
   printf(" The sum of %ld and %ld  is %ld\n\n", fno, sno, sum);
   return 0;
}
long addTwoNumbers(long *n1, long *n2)
{
   long sum;
   sum = *n1 + *n2;
   return sum;
}
```

```
Pointer : Add two numbers using call by reference:
-------------------------------------------------------
 Input the first number : 5
 Input the second  number : 6
 The sum of 5 and 6  is 11
```

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

Test Data :

Input the first number : 5

Input the second number : 6

*Expected Output* :

```
6 is the maximum number.
```

```
#include <stdio.h>
#include <stdlib.h>
void 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);
 }

}
```

Copy

Sample Output:

```
 Pointer : Find the maximum number between two numbers :
------------------------------------------------------------
 Input the first number : 5
 Input the second  number : 6

 6 is the maximum number.
```

4. Write a program in C to store n elements in an array and print the elements using pointer.

Test Data :

Input the number of elements to store in the array :5

Input 5 number of elements in the array :

element - 0 : 5

element - 1 : 7

element - 2 : 2

element - 3 : 9

element - 4 : 8

*Expected Output* :

```
 The elements you entered are :
 element - 0 : 5
 element - 1 : 7
 element - 2 : 2
 element - 3 : 9
 element - 4 : 8
```

```c
#include <stdio.h>
int main()
{
   int arr1[25], i,n;
   printf("\n\n Pointer : Store and retrieve elements from an array :\n");
   printf("------------------------------------------------------------\n");
   printf(" Input the number of elements to store in the array :");
   scanf("%d",&n);

   printf(" Input %d number of elements in the array :\n",n);
   for(i=0;i<n;i++)
      {
          printf(" element - %d : ",i);
          scanf("%d",arr1+i);
          }
   printf(" The elements you entered are : \n");
   for(i=0;i<n;i++)
      {
          printf(" element - %d : %d \n",i,*(arr1+i));
          }
           return 0;
}
```

Copy

Sample Output:

```
 Pointer : Store and retrieve elements from an array :
------------------------------------------------------------
 Input the number of elements to store in the array :5
 Input 5 number of elements in the array :
 element - 0 : 5
 element - 1 : 7
 element - 2 : 2
 element - 3 : 9
 element - 4 : 8
 The elements you entered are :
 element - 0 : 5
 element - 1 : 7
 element - 2 : 2
 element - 3 : 9
 element - 4 : 8
```

5. Write a program in C to print all permutations of a given string using pointers.

*Expected Output* :

```
The permutations of the string are :
abcd  abdc  acbd  acdb  adcb  adbc  bacd  badc  bcad  bcda
bdca  bdac  cbad  cbda  cabd  cadb  cdab  cdba  db
ca  dbac  dcba  dcab  dacb  dabc
```

```c
#include <stdio.h>
#include <string.h>

void changePosition(char *ch1, char *ch2)
{
    char tmp;
    tmp = *ch1;
    *ch1 = *ch2;
    *ch2 = tmp;
}
void charPermu(char *cht, int stno, int endno)
{
   int i;
   if (stno == endno)
     printf("%s  ", cht);
   else
   {
       for (i = stno; i <= endno; i++)
       {
          changePosition((cht+stno), (cht+i));
          charPermu(cht, stno+1, endno);
          changePosition((cht+stno), (cht+i));
       }
   }
}

int main()
{
    char str[] = "abcd";
   printf("\n\n Pointer : Generate permutations of a given string :\n");
   printf("--------------------------------------------------------\n");
    int n = strlen(str);
    printf(" The permutations of the string are : \n");
    charPermu(str, 0, n-1);
     printf("\n\n");
    return 0;
}
```

Sample Output:

```
 Pointer : Generate permutations of a given string :
--------------------------------------------------------
 The permutations of the string are :
```

```
abcd   abdc   acbd   acdb   adcb   adbc   bacd   badc   bcad   bcda
bdca   bdac   cbad   cbda   cabd   cadb   cdab   cdba   db
ca   dbac   dcba   dcab   dacb   dabc
```

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

Test Data :

Input a string : Sini Raj P

*Expected Output* :

```
The length of the given string w3resource
is : 10
```

```
#include <stdio.h>
int calculateLength(char*);

void 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");

}

int calculateLength(char* ch) // ch = base address of array str1 ( &str1[0]  )
{
   int ctr = 0;
   while (*ch != '\0')
   {
      ctr++;
      ch++;
   }
   return ctr;
}
```

Copy

Sample Output:

```
 Pointer : Calculate the length of the string :
---------------------------------------------------
```

```
Input a string :Sini Raj P
The length of the given string w3resource
is : 10
```

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

Test Data :

Input the value of 1st element : 5

Input the value of 2nd element : 6

Input the value of 3rd element : 7

*Expected Output* :

```
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
```

```c
#include <stdio.h>
void swapNumbers(int *x,int *y,int *z);
int 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);
    return 0;
}
void swapNumbers(int *x,int *y,int *z)
{
    int tmp;
    tmp=*y;
    *y=*x;
    *x=*z;
    *z=tmp;
}
```

Copy

Sample Output:

```
 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
```

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

Test Data :

Input a number : 5

*Expected Output* :

```
The Factorial of 5 is : 120
#include <stdio.h>
void findFact(int,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);
         return 0;
        }

void findFact(int n,int *f)
                {
        int i;
        *f =1;
        for(i=1;i<=n;i++)
        *f=*f*i;
        }
```

Copy

Sample Output:

```
 Pointer : Find the factorial of a given number :                       
------------------------------------------------------                  
 Input a number : 5                                                     
 The Factorial of 5 is : 120
```

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

Test Data :

Input a string: string

*Expected Output* :

```
Number of vowels : 1                                                   
Number of constant : 5
```

```
#include <stdio.h>
int 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: ");
    fgets(str1, sizeof str1, stdin);

    //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);
    return 0;
}
```

Copy

Sample Output:

```
Pointer : Count the number of vowels and consonants :
----------------------------------------------------------
 Input a string: string
 Number of vowels : 1
 Number of consonants : 5
```

10. Write a program in C to sort an array using Pointer.

Test Data :

testdata

*Expected Output* :

Test Data :
Input the number of elements to store in the array : 5
Input 5 number of elements in the array :
element - 1 : 25
element - 2 : 45
element - 3 : 89
element - 4 : 15
element - 5 : 82
*Expected Output* :

```
 The elements in the array after sorting :
 element - 1 : 15
 element - 2 : 25
 element - 3 : 45
 element - 4 : 82
 element - 5 : 89
```

```
#include <stdio.h>
void main()
{
   int *a,i,j,tmp,n;
        printf("\n\n Pointer : Sort an array using pointer :\n");
        printf("--------------------------------------------\n");

   printf(" Input the number of elements to store in the array : ");
   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",a+i);
          }
   for(i=0;i<n;i++)
   {
    for(j=i+1;j<n;j++)
    {
       if( *(a+i) > *(a+j))
       {
      tmp = *(a+i);
      *(a+i) = *(a+j);
      *(a+j) = tmp;
       }
    }
   }
   printf("\n The elements in the array after sorting : \n");
   for(i=0;i<n;i++)
      {
          printf(" element - %d : %d \n",i+1,*(a+i));
          }
printf("\n");
}
```

Copy

Sample Output:

```
Pointer : Sort an array using pointer :
--------------------------------------------
 Input the number of elements to store in the array : 5
 Input 5 number of elements in the array :
 element - 1 : 25
 element - 2 : 45
 element - 3 : 89
 element - 4 : 15
 element - 5 : 82

 The elements in the array after sorting :
 element - 1 : 15
 element - 2 : 25
 element - 3 : 45
 element - 4 : 82
 element - 5 : 89
```

11. Write a program in C to show how a function returning pointer.

Test Data :

Input the first number : 5

Input the second number : 6

*Expected Output* :

```
 The number 6 is larger.
```

```
#include <stdio.h>
int* findLarger(int*, int*);
void main()
{
 int numa=0;
 int numb=0;
 int *result;
        printf("\n\n Pointer : Show a function returning pointer :\n");
        printf("--------------------------------------------------\n");
   printf(" Input the first number : ");
   scanf("%d", &numa);
   printf(" Input the second  number : ");
   scanf("%d", &numb);

 result=findLarger(&numa, &numb);
 printf(" The number %d is larger.  \n\n",*result);
}

int* findLarger(int *n1, int *n2)
{
 if(*n1 > *n2)
  return n1;
 else
  return n2;
}
```

Copy

Sample Output:

```
 Pointer : Show a function returning pointer :
--------------------------------------------------
 Input the first number : 5
 Input the second  number : 6
 The number 6 is larger.
```

12. Write a program in C to compute the sum of all elements in an array using pointers.  Test Data :

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

*Expected Output* :

```
 The sum of array is : 20
```

```
#include <stdio.h>
void 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; // pt store the base address of array arr1

   for (i = 0; i < n; i++) {
      sum = sum + *pt;
      pt++;
   }

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

Sample 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
```

13. Write a program in C to print the elements of an array in reverse order.

Test Data :

Input the number of elements to store in the array (max 15) : 5

Input 5 number of elements in the array :

element - 1 : 2

element - 2 : 3

element - 3 : 4

element - 4 : 5

element - 5 : 6

*Expected Output* :

```
 The elements of array in reverse order are :
 element - 5 : 6
 element - 4 : 5
 element - 3 : 4
 element - 2 : 3
 element - 1 : 2
```

```
#include <stdio.h>
void main()
{
   int n, i, arr1[15];
   int *pt;
        printf("\n\n Pointer : Print the elements of an array in reverse order
:\n");
        printf("----------------------------------------------------------------
\n");

   printf(" Input the number of elements to store in the array (max 15) : ");
   scanf("%d",&n);
   pt = &arr1[0];  // pt stores the address of base array arr1
   printf(" Input %d number of elements in the array : \n",n);
   for(i=0;i<n;i++)
      {
          printf(" element - %d : ",i+1);
          scanf("%d",pt);//accept the address of the value
          pt++;
          }

   pt = &arr1[n - 1];

   printf("\n The elements of array in reverse order are :");

   for (i = n; i > 0; i--)
   {
      printf("\n element - %d : %d  ", i, *pt);
      pt--;
   }
printf("\n\n");
}
```

Sample Output:

```
 Pointer : Print the elements of an array in reverse order :
--------------------------------------------------------------
--
 Input the number of elements to store in the array (max 15) :
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 elements of array in reverse order are :
 element - 5 : 6
 element - 4 : 5
 element - 3 : 4
 element - 2 : 3
 element - 1 : 2
```

14. Write a program in C to print all the alphabets using pointer

```
#include <stdio.h>

int main()
{
    char alph[27];
    int x;
    char *ptr;
        printf("\n\n Pointer : Print all the alphabets:\n");
        printf("----------------------------------------\n");
    ptr = alph;

    for(x=0;x<26;x++)
    {
        *ptr=x+'A';
        ptr++;
    }
    ptr = alph;

printf(" The Alphabets are : \n");
    for(x=0;x<26;x++)
    {
       printf(" %c ", *ptr);
        ptr++;
    }
    printf("\n\n");
    return(0);
}
```

Copy

Sample Output:

```
 Pointer : Print all the alphabates:
----------------------------------------
 The Alphabets are :
 A  B  C  D  E  F  G  H  I  J  K  L  M  N  O  P  Q  R  S  T  U
V  W  X  Y  Z
```

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

Test Data :

Input a string : w3resource

*Expected Output* :

```
 Pointer : Print a string in reverse order :
------------------------------------------------
 Input a string : SINI
 Reverse of the string is : INIS
```

```
#include <stdio.h>
int 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);
    return 0;
}
```

## Sample Output:

```
 Pointer : Print a string in reverse order :
------------------------------------------------
 Input a string : w3resource
 Reverse of the string is : ecruoser3w
```