1)

```
#include <stdio.h>
void main(void)
{
int m=10,n,o;
int *z=&m ;

        printf("\n\n Pointer : Show the basic declaration of pointer :\n");
        printf("-------------------------------------------------------\n");
        printf(" Here is m=10, n and o are two integer variable and *z is an
integer");
        printf("\n\n z stores the address of m  = %p\n",  z); // z is a pointer
so %p would print the address
        printf("\n *z stores the value of m = %i\n",   *z);
        printf("\n &m is the address of m = %p\n",   &m); // &m gives the
address of the integer variable m
                        // so %p is the specifier for that address
        printf("\n &n stores the address of n = %p\n",   &n);
        printf("\n &o  stores the address of o = %p\n",   &o);
        printf("\n &z stores the address of z = %p\n\n", &z); // &z gives
the address, where the pointer z is
                        // stored -> still an address -> %p is the right
                        // specifier
}
```

2)

```
#include <stdio.h>
int main()
{
  int* ab;
```

```
  int m;
  m=29;
   printf("\n\n Pointer : How to  handle the pointers in the program :\n");
   printf("------------------------------------------------------------\n");
   printf(" Here in the declaration ab = int pointer, int m= 29\n\n");

  printf(" Address of m : %p\n",&m);
  printf(" Value of m : %d\n\n",m);
  ab=&m;
  printf(" Now ab is assigned with the address of m.\n");
  printf(" Address of pointer ab : %p\n",ab);
  printf(" Content of pointer ab : %d\n\n",*ab);
  m=34;
  printf(" The value of m assigned to 34 now.\n");
  printf(" Address of pointer ab : %p\n",ab);
  printf(" Content of pointer ab : %d\n\n",*ab);
  *ab=7;
  printf(" The pointer variable ab is assigned the value 7 now.\n");
  printf(" Address of m : %p\n",&m);//as ab contain the address of m
                        //so *ab changed the value of m and now m
become 7
  printf(" Value of m : %d\n\n",m);
  return 0;
}


3)
#include <stdio.h>
void main()
{
 int m=300;
```

```
float fx = 300.60;
char cht = 'z';

 printf("\n\n Pointer : Demonstrate the use of & and * operator :\n");
 printf("-----------------------------------------------------\n");


int *pt1;
float *pt2;
char *pt3;
pt1= &m;
pt2=&fx;
pt3=&cht;
printf ( " m = %d\n",m);
printf ( " fx = %f\n",fx);
printf ( " cht = %c\n",cht);
printf("\n Using & operator :\n");
printf("----------------------\n");
printf ( " address of m = %p\n",&m);
printf ( " address of fx = %p\n",&fx);
printf ( " address of cht = %p\n",&cht);
printf("\n Using & and * operator :\n");
printf("---------------------------\n");
printf ( " value at address of m = %d\n",*(&m));
printf ( " value at address of fx = %f\n",*(&fx));
printf ( " value at address of cht = %c\n",*(&cht));
printf("\n Using only pointer variable :\n");
printf("--------------------------------\n");
printf ( " address of m = %p\n",pt1);
printf ( " address of fx = %p\n",pt2);
```

```
  printf ( " address of cht = %p\n",pt3);
  printf("\n Using only pointer operator :\n");
  printf("--------------------------------\n");
  printf ( " value at address of m = %d\n",*pt1);
  printf ( " value at address of fx= %f\n",*pt2);
  printf ( " value at address of cht= %c\n\n",*pt3);
}
```

4)

```
#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;
```

```
}
```

5)

```
#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;
}
```

6)

```
#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);
 }

}
```

```
7) #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;
}
```

8)

```
#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;
}
```

9)

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

```
#include <stdlib.h>
int main()
{
   int i,n;
   float *element;
        printf("\n\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));  // Memory is allocated for 'n'
elements
   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);
```

```
   return 0;
}
```

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;
}
```

11)

```
#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;
}
```

12)

```
#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;
    }
```

13)

```
#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;
}
```

14)

```
#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");
}
```

15)

```
#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;
}
```

16)

```
#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);
}
```

17)

```
#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");
}
```

18)

```
#include <stdio.h>
struct EmpAddress
{
char *ename;
char stname[20];
int pincode;
}
employee={"John Alter","Court Street \n",654134},*pt=&employee;

int main()
{
```

```
        printf("\n\n Pointer : Show the usage of pointer to structure :\n");
        printf("-----------------------------------------------------\n");
        printf(" %s from %s \n\n",pt->ename,(*pt).stname);
        return 0;
}
```

19)

```
#include <stdio.h>
union empAdd
{
char *ename;
char stname[20];
int pincode;
};

int main()
{
        printf("\n\n Pointer : Show a pointer to union :\n");
        printf("-------------------------------------\n");
   union empAdd employee,*pt;
   employee.ename="Jhon Mc\0Donald";//assign  the string up to null character i.e. '\0'

   pt=&employee;

   printf(" %s %s\n\n",pt->ename,(*pt).ename);

   return 0;
}
```

20)

```
#include <stdio.h>
struct employee
{
char *empname;
int empid;
};

int main()
{
        printf("\n\n Pointer : Show a pointer to an array which contents are
pointer to structure :\n");
        printf("-------------------------------------------------------------------
----------\n");

        static struct employee
emp1={"Jhon",1001},emp2={"Alex",1002},emp3={"Taylor",1003};
        struct employee(*arr[])={&emp1,&emp2,&emp3};
        struct employee(*(*pt)[3])=&arr;

        printf(" Exmployee Name : %s \n",(**(*pt+1)).empname);
        printf("---------------- Explanation --------------------\n");
        printf("(**(*pt+1)).empname\n");
        printf("= (**(*&arr+1)).empname   as pt=&arr\n");
        printf("= (**(arr+1)).empname     from rule *&pt = pt\n");
        printf("= (*arr[1]).empname       from rule *(pt+i) = pt[i]\n");
        printf("= (*&emp2).empname        as arr[1] = &emp2\n");
        printf("= e2.empname = Alex       from rule *&pt = pt\n\n");
        printf(" Employee ID :  %d\n",(*(*pt+1))->empid);
        printf("---------------- Explanation --------------------\n");
```

```
        printf("(*(*pt+1))-> empid\n");
        printf("= (**(*pt+1)).empid     from rule -> = (*).\n");
        printf("= emp2.empid = 1002\n");
        printf("\n\n");
        return 0;
}
```

21)

```
#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);
}
```

22)

```
#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;
}
```