Assignment

1.
```
# include <stdio.h>
void fun (int x)
 { x = 30 ;
 }
int main()
{
 int y = 20;
 fun (y);
 printf ("%d, y);
  return 0;
 }
```

Output : 20

The paramete is passed by value the value of y is not modified using function.

The value changes in the function

The y value remains same as original value.

2)
```
# include<stdio.h>
void fun (int *ptr)
 {
   *ptr = 30;
 }
 int main()
  {
   int y=20;
   fun( &y) ;
   printf ( "%d, "y);
     return 0 ;
    }
```

Output : 30

The paralme is passed by reference

The value of the address ptr is changed to 30.

The address ptr is modified by the function.

3)
```
Int main ()
{
 int *ptr ;
  int x ;
  ptr = &x;                              • ptr points to x
  *ptr = 0;                                value is 0
  printf ("x = %d\n',x);                      x = 0
  printf (" *ptr = %d\n,' *ptr);        ptr value = 0
    *ptr += 5 ;                          increment ptr by 5
   printf ("x= %.d\n", x);                   x = 5
   Print f (*ptr= %d\n",*ptr);          ptr value = 5
    (*ptr) ++ ;                         increment ptr by 1
printf("x= %d \n", x);
printf ("* ptr = %d\n ,* ptr);
  return 0;
  }
```

x = 6

ptr value is 6

x = 0

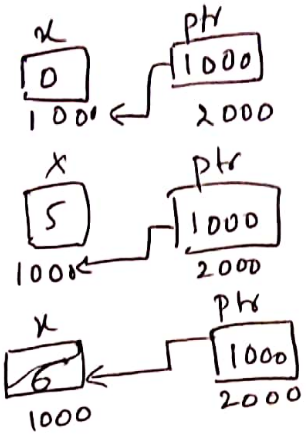
*ptr = 0

x = 5

* ptr = 5

x = 6

*ptr = 6



```
4) # include <stdio.h>
  int main ()
  {
  char  s1[7] = "1234";
      p = s1 +2
        *p = '0' ;
      printf (" %s", s1) ;
      }
```

s1 is the first characte of array , p will hold the address of the 3rd character

The value at p is chaged to 0

Output :- 1204

```
5) #include <stdio.h>
   void f( int *p , int *q)
     {
      p= q ;
      *p=2
     }
    int i=0 ; j=1 ;
     int main ()
     {
      f (&i ,&j) ;
     printf ("%d %d \n", i, j);
       getchar ();
       return 0;
     }
```

output : 0 2

p to address of i

q to address of j

p= q

the p points to address of j

*p=2 the value of j = 2

sprints i=0
j=2

6)
```
# include <stdio.h>
int f ( int x, int * py, int ** pp2)

 {
 int y, z ;
  x * ppz ;
  * py+ =2 ;
  y = * py;
   x+  = 3;
  rehuen x + y+ z ;
  }
 Void main ()
 { int c , * b , x * a;
   c = 4
    b=&c
   a= &b
   print f( "%d", f( c, b, a);
       rehuen 0;
     }.
```

c = 4 + 1 = 5

z = 5

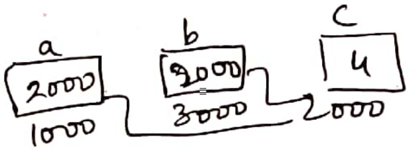
c = 5+2 = 7

y = 7

x = 4 + 3 = 7

$\longrightarrow$ 7 + 7 + 5 = 19.

output= 19



7)
```
# include <Stdio.h>
 int main( )
 { int arr[] ={ 1, 2, 3, 4, 5 };
    int *p= arr ;
   ++ * p ;
   p+ =2;
   print f ( "%.d" , *p) ;
    rehuen 0;
   }
```

first clemet of array is pointed by p.

++ * p is evaluated as ++(* p)

The first element value will be incrmented by 1.

In p+ =2 , the p value is changed to 3 element address.

Output : 3.

```
8) # include < stdio.h>
   int main()
   {
   char c[] = "GATE2011";
   char * p=c
   printf ( "%s", p+p[3] - p[1]);
   }.
```

p[3]= E

p [1]= A

E - A = 69 - 65 = 4 .

p+4. = 2011

(ans address of p+4)

Output : 2011 .

```
9) int main()
   {
   char arr[] = "workstreet";
   printf ("%s", arr + 4);
   reten o;
   }
```

The arr+4 is address of the string 'street'.

output = Street .

```
10) # include<stdio.h>
   int fun (char *str1)
   {
   char *str2 =str1;
   while (* ++ str1);
   return (str1 -str2);
   }
   int main()
   {
   char * str ="workstreet";
   printf ("%d", fun (str));
   rehun o;
   }.
```

output =10

*str2 is initialized to str1

str1 is incremented by 10

str1 - str2 = 10

The fun counts the number of charcter in the string