Unit - 4

Pointers
Union
Dynamic memory (Malloc / Calloc / Realloc)
File → Creat / Write / Read
Common Line Arguments.

Pointers:

A pointer is a variable whose value is address of another variable

Syntax:

data type * variable name;

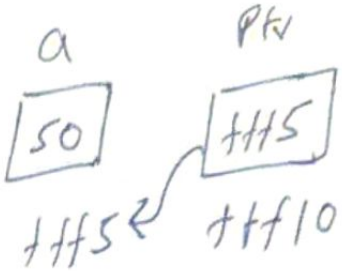
Exam

```
int * a;
char * c;
float *f;
```

```
#include <stdio.h>
int main()
{
int a = 50;
int * ptr; // declaring pointer variable.
ptr = &a; // value assignment
printf("%d", ptr)
```

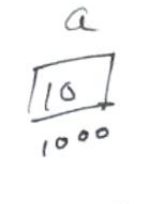
```
printf("%d", a);      ⟶ 50
printf("%d", ptr);    ⟶ fff5
printf("%d", *ptr);   ⟶ 50

return 0;
```

3



Pointers.

```
void main()
{
    int a = 10;
    int * ptr;
    ptr = &a;
    printf("%d", a);      ⟶ 10
    printf("%d", &a);     ⟶ 1000
    printf("%d", ptr);    ⟶ 100
    printf("%d", &ptr);   ⟶ 2000
    Printf("%d", *ptr);   ⟶ 10
```




```
X ptr = a;
```

→ can't assign int to int*

& → address

* → value at address.
* pointer.

%d

Types:

1) Typed pointer

```
int *p;
char *c;
```

2) Untyped pointer / generic pointer

```
void *ptr;
```

Size of Pointer.

```
void main()
{ int *i;
  char c;
  float *f;
  double *d;
  printf("%d", sizeof(i));   → 2
  printf("%d", sizeof(c));   → 1
  printf("%d", sizeof(f));   → 2
  printf("%d", sizeof(d));   → 2
```

check for

```
i++;
f++;
*c
d = c + i
d = f + i
```

```
f = f + i;
f = d - i;
f = f + 4;
f = i - 3;
```

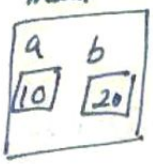
```
f = f * 4;
d = d / 5;
}
```

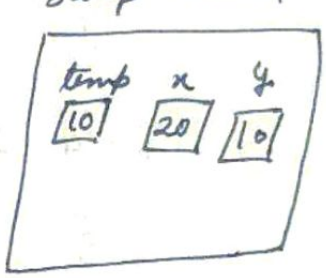
Arithmetic Operation of pointers:-

Perform the operations on last page.

Call by Value:-

```
Void main ()
{
  int a=10, b=20;
  printf ("Before swapping: a=%d and b=%d", a, b);
  swap (a, b);
}
Void swap (int x, int y)
{
  int temp=x;
  x=y;
  y=temp;
  printf ("The value of a is %d and b is %d", x, y);
}
printf ("After swapping; a=%d, b=%d", a, b);
}
```



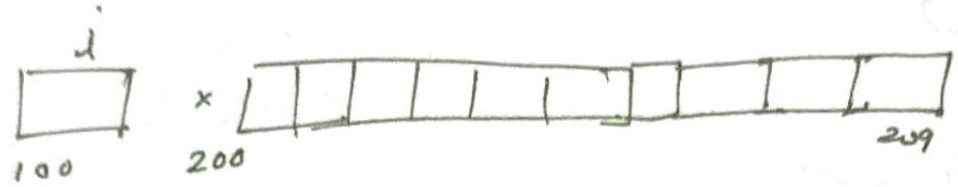


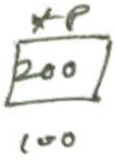
Call by reference:

```
Void main()
{ int a=10 , b=20;
  printf ("Before Swapping : a=%d and b=%d", a,b);
  swap(&a, &b);
  printf ("After swapping a=%d , b=%d" a, b );
  }
  Void swap (int*x , int*y)
  {
    int temb = *x;
    *x = *y;
    *y = temb;
    }.
```

Q^n int palindrom (int*);

Pointer to array :-






```
#include <stdio.h>
void main()
{
int i, x[10], *P;
P = X;        → base address.
printf("Enter the elements in an array");
for(i=0; i<10; i++)
scanf("%d", (P+i));      & P[i]

for (i=0; i<10 ; i++)
printf("%d", *(P+i));
```

```
#include <stdio.h>
void main()
{
int i, x[10], *p;
   p = x;
printf("Enter the elements in an Array");
for(i=0; i<10; i++)
      scanf("%d", p);
        p++;
   }
   p = x;
for (i= 0; i<10; i++)
   {
     printf("%d", *p);
       p++;
   }
}
```

address

to print.

Q. W.A.P. to calculate avg of elements of an array using function.

```
#include <stdio.h>
void main ()
{
int i , x[10] , *p;
float Avg (int [], int);        ] → f^n prototype
   p= x;
printf("Enter the elements in an array");
for (i=0;  i<10 ; i++)
    {scanf ("%d", &x[i]);
    }
printf ("the average is %f ", Avg (x,10));     → f^n call
}.
float Avg (int p[], int n)
  {int sum=0, i;
  for (i=0 ; i<10; i++)
  {
    sum = sum + P[i];
  }
  return( sum/10.0);
  }.
```

Pointer to SDS (Single Dimension String)

```
#include <stdio.h>
void main ()
{
  char ch[50], *c;
    c=ch;
   gets(c);
    puts(c);
}
```

Qn W.A.P to check whether given string is palindrome or not.

& assign index to each character in

Qn WAP to check whether given strin using function

Qn WAP to check whether given string is palindrome or not without using library function.

(*) ~~How~~ Array of Pointer : Array of pointer.

```
Void main ()
{
 int X = 10; Y=20; Z =30;
 int * P[3];
 P[0] = &X;
 P[1] = &Y;
  P[2]=&Z;
printf ("X=%d, Y=%d, Z=%d," *P[0], *P[1], *P[2]);
}
```