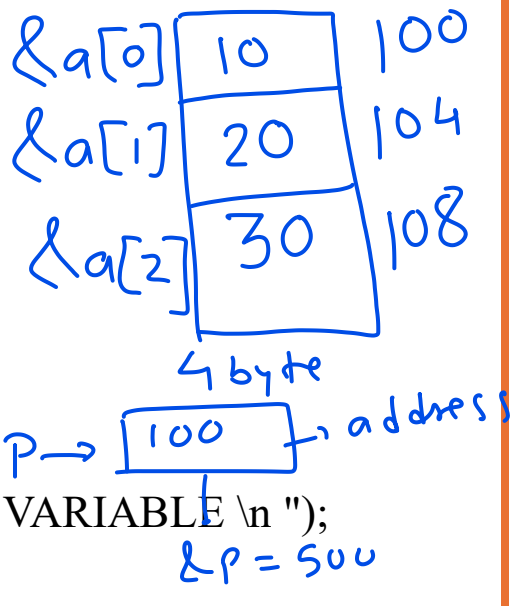
// POINTER AND ARRAY

// WAP TO PRINT N NOS USING POINTER.



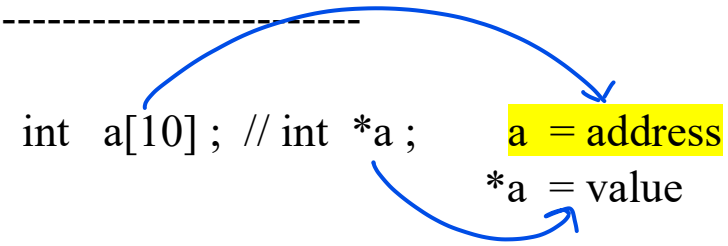

```
#include<stdio.h>
      int    main()
      {
            float  a[] = { 10, 20, 30 } ;
            int  i , n = 3 ;
            float  *p ;

            p = &a[0];
 printf(" PRINT NOS USING POINTER VARIABLE \n ");
           for(  i = 0 ; i < n ; i++ )
           {
                 printf(" %u \t %f \n " , p , *p);

              p++ // NEXT ADDRESS ( LOCATION )
           }
}
```

---

n = 3 p = 100
for i = 0 to 2
i=0 p = 100 *p =10
i = 1 p = 104 *p =20
i = 2 p = 108 *p = 30

i = 3 p = 112 X

ONE DIM. ARRAY



&a[i] = (a+i) // address

a[i] = *(a+i) // value

base address = a = (a + 0) = &a[0]

---

int a[10]; // DECLARATION

&i[a] or &3[a] // PROCESSING

&i[a] = (i + a)

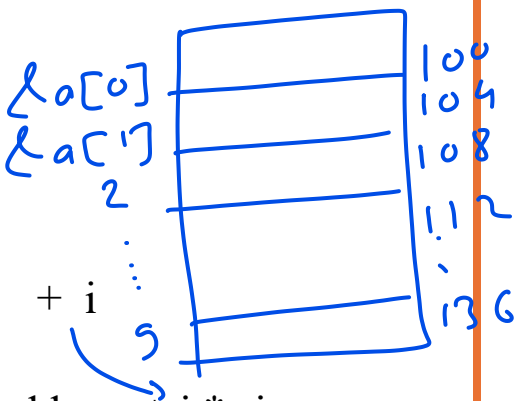
i[a] = *(i + a)

---

float a[10];

base address = 100

FIND ADDRESS OF &a[3]



e.g. i = 3 , size = 4 byte

ADDRESS OF &a[i] = a + i

ADDRESS OF &a[i] = base address + i * size

ADDRESS OF &a[3] = 100 + 3 * 4

= 112

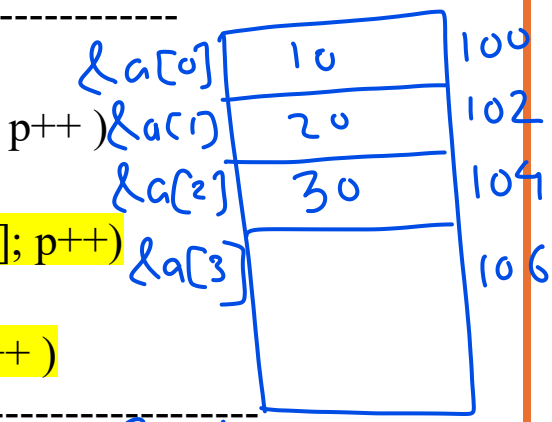
POINTER AND STRING

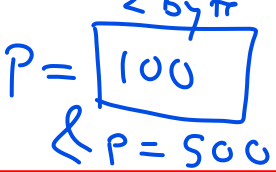
```
char   a[10];   // char  *a ;

        base   address = a ;

        input  string -->  scanf("%s", a);
```

// INPUT AND PRINT N NOS USING ARRAY

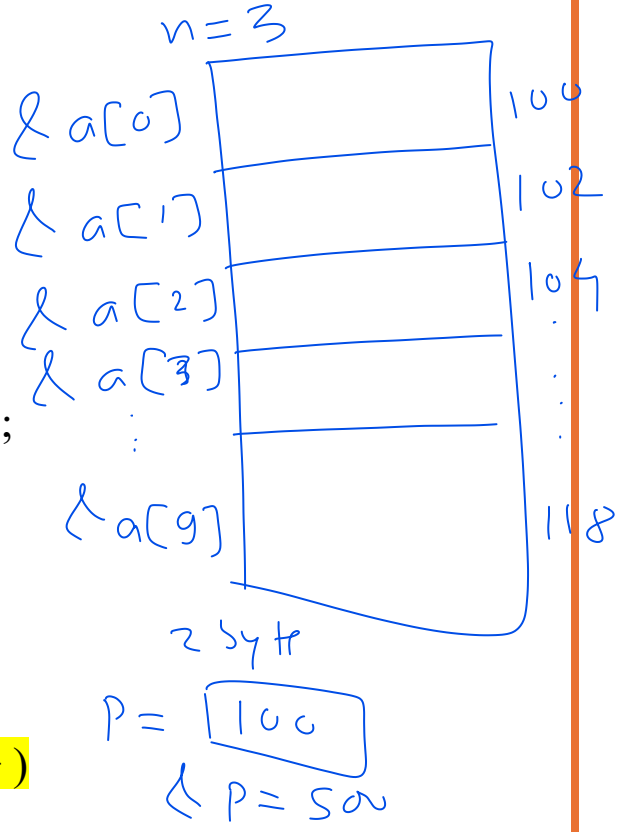
```
#include<stdio.h>
    int    main()
    {
        int  a[10] , i , n ;
```

```
        printf(" ENTER SIZE \n ");
        scanf("%d", &n);
         printf(" ENTER NOS \n ");
         for( i = 0 ; i < n ; i++)
          {
                scanf("%d", (a+i) );  // &a[i]= (a+i) // address
          }
          printf(" NOS = \n");
         for( i = 0 ; i < n ; i++ )
          {
               printf(" %d\n" ,*(a+i) );  // a[i] = *(a+i)      // value
          }
   }

/*
-------------------------------------------------------------------------
        LOOP  THROUGH  POINTER  VARIABLE
          ---------------------------------------------
                n = 3
                for( p = 100 ;  p  < 106  ; p++ )
                      or
               for( p = &a[0] ; p < &a[n]; p++)
                      or
               for( p = a ; p < (a+n) ; p++ )
 ---------------------------------------------------------------------
*/
```



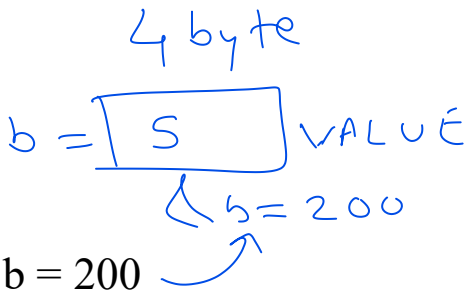


WAP TO INPUT AND PRINT NOS USING LOOP THROUGH POINTOR




```
#include<stdio.h>

    int   main()
    {
          int  a[10] , n , *p ;

         printf(" ENTER SIZE \n ");

          scanf("%d" , &n );

          printf(" ENTER NO \n " );

          for( p = a ; p < (a+n) ; p++ )
          {
                scanf("%d", p ) ; // address
          }

          printf(" NO \n " );

          for( p = a ; p < (a+n) ; p++ )
          {
                printf(" %d\n " , *p ) ; // value
          }
 }
```

```
#include<stdio.h>
int   main()
{
      int b= 5;   // normal variable

      printf("%d\n",b);        // 5

      printf("%u\n",&b);   // address of b = 200

      &b++ ;  // can't change base address

      int   a[10];  // array  --> base address a = &a[0] =100

      a++ ;  // can't change base addres

      int  *p; // pointor varible

      p = a ; or p = &a[0] ; // p = 100

      p++;    // p = 102

}
```



POINTOR AIRTHMATICS

float *p ; // return address

p = &a[3] = 112 ; p++ -----> 116

p = &a[3] = 112 ; p-- ------> 108

p = &a[3] = 112 ; p + 2 ----->

p + 2 (constant addition)

p = &a[3] = 112 ; p - 2 ----->

p - 2 (constant substraction)

---

float *p ; // return address

p = &a[3] = 112 ; p++ -----> 116

p = &a[3] = 112 ; p-- -----> 108

p = &a[3] = 112 ; p + 2 -----> 112 + 2 * 4 = 120

p + 2 (constant addition)

p = &a[3] = 112 ; p - 2 -----> 112 - 2 * 4 = 104

p - 2 (constant substraction)

---

p = &a[3] = 112

p = &a[6] = 124

q - p = return int value

= 124 - 112 = 12 byte

difference = 12 / 4 (SIZE OF THE DATA TYPE)

= 3 LOCATION

```
// WAP PROGRAM TO FIND BIGGEST NO. USING  POINTOR
#include<stdio.h>
int  main()
{
   int  a[10],n,b,*p;

   printf("enter size\n");
   scanf("%d",&n);

   printf("enter no\n");
   for(p = a ; p <(a+n) ; p++)
   {
      scanf("%d",p);
   }

   b=a[0];

   for(p = a ; p <(a+n); p++)
   {
      if(b < *p)
      {
b=*p;
      }
   }
   printf("biggest no=%d\n",b);
}
```

ARRAY OF POINTER

```
#include<stdio.h>

  int    main()
  {
            int  a = 2 , b = 3 , c = 4 , i ;

            int  *p[10];   //  ARRAY OF POINTER

             p[0] = &a ;

             p[1] = &b ;

             p[2] = &c ;

            for( i = 0 ; i < 3 ; i++ )
             {
                   printf("%d \t", *p[i]);  //  2     3     4
             }

  }
```

INPUT AND PRINT MATRIX

```
#include<stdio.h>

int   main()
{
        int  a[10][10] , i , j , m , n ;

       printf(" ENTER ROW AND COL \n " );

        scanf("%d%d" , &m , &n);

      printf(" ENTER MATRIX  ELEMENTS \n ");

       for( i = 0 ; i < m ; i++ )     // ROWS
        {
              for( j = 0 ; j < n ; j++ )   // COLS
               {

      scanf("%d", *(a+i)+j);    // &a[i][j] = (a[i]+j) = *(a+i) +j

               }

       } // INPUT MATRIX
```

int a[10][10]; // int **a;

&a[i][j] = (a[i]+j) = (*(a+i)+j) //address

a[i][j] = *(*(a+i)+j) //value

```
            printf(" MATRIX  A \n");

            for( i = 0 ; i < m ; i++)
             {
                    for( j = 0 ; j < n ; j++)
                     {

   printf("%4d",*(*(a+i)+j)); // a[i][j] = *(a[i]+j) = *(*(a+i)+j)

                     }

                     printf("\n");

            }  // PRINT MATRIX
      }

// INPUT AND PRINT MATRIX  USING POINTOR VARIABLE

#include<stdio.h>
      int   main()
      {
             int  a[10][10] , i , j , m , n ;
             int *p;

            printf(" ENTER ROW AND COL \n " );
```

```
   scanf("%d%d" , &m , &n);
printf(" ENTER MATRIX  ELEMENTS \n ");
   for( i = 0 ; i < m ; i++ )     // ROWS
   {
            p = a[i] ; // copy row address

         for( j = 0 ; j < n ; j++ )   // COLS
          {
                  scanf("%d", (p+j));
          }
   } // INPUT MATRIX

   printf(" MATRIX  A \n");

  for( i = 0 ; i < m ; i++)
   {
      p =a[i];

          for( j = 0 ; j < n ; j++)
           {
                   printf("%4d",*(p+j));
           }
            printf("\n");

  }  // PRINT MATRIX
}
```