1.int \*p[10];

2.printf("enter 5 element\n");

**scanf("%d",&i[p]); //pointer**

You can not write (“%d”, p[i] ) bz we take value in array as **scanf(“%d”,&a[i]) // Array**

we use **printf("%d\n",a[i])** to print array so similarly we use

**printf("%d\n",p[i])// Print Pointer**

OR

**scanf("%d",p+i)**

OR

**scanf("%d",(p+i))//** No need ‘&’

BUT scanf("%d",&(p+i)) this not possible ,error.

3. **if(i[p]>j[p]) // Pointer**

OR

**if(a[i]>a[j])// Array**

OR

**If(\*(p+i)>\*(p+j))//Pointer**

4. **printf("%d\n",i[p])**

OR

**printf("%d\n",a[i])**

OR

**Printf(“%d”,\*(p+i));**

We can not use printf(“%d”,\*p+i) we need bracket like **\*(p+i).**

1.

int a[10];

int \*p=a;

2. scanf("%d",&a[i]);

3. printf("%d\n",i[p]);

4. if(p[i] > p[j])

{ int temp=i[p];

i[p]=j[p];

j[p]=temp;

}

OR

if(\*(p+i)>\*(p+j))

{ int temp=\*(p+i);

\*(p+i)=\*(p+j);

\*(p+j)=temp; }

**Important think:**

**Case 1: we directly pointed to *any* l*ocation* of array so we need ‘&’ .**

Int a[]={1,2,3,4,5};

Int \*p;

P=&a[3];

char str[]="akashthakare";

char \*ptr=&str[4];

**Case 2:First we pointed main array then pointed any location using \*p then no need to use &**

**Otherwise we need & .**

Int a[]={1,2,3,4,5};

Int \*p=a;

**\*P=a[3]; OR p=&a[3];**

char str[]="akashthakare";

char \*ptr=str;

\*ptr=str[4]; OR ptr=&str[4]

**Difference between ++\*p, \*p++ and \*++p.**

int main()

{

int a[]={11,21,31,41};

int \*p;

p=a;

//value of p incremented by 1 and returned

/\*printf("%d ",++\*p);//12

printf("%d ",\*p);//12\*/

//value of p is returned pointer incremented by 1 //see:here value not increament

/\*printf("%d ",\*p++);//11

printf("%d ",\*p);//21\*/

//pointer incremented by 1 value returned

printf("%d ",\*++p);//21

printf("%d ",\*p);//21

**Void pointer(Generic Pointer):** is a pointer that point variable of any data type;

**Type coversion required:**

int a[]={1,2,3,4,5};

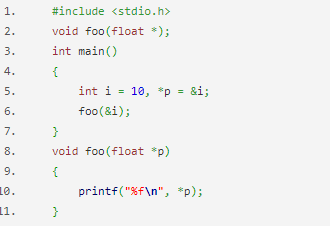
void \*p;

p=a;

printf("%d",\*((int \*)p));// **This is type casting for** **Void pointer**

OR

Printf(“%d”,\*(int \*)p);



Printf(“%f”,(float)\*p); // **this type casting for type conversion.**

**Null Pointer** : is a pointer which points nothing

char a[]="akashthakare";

char \*p=NULL;

printf("%d",p);// zero(0)

p=&a[5];

printf("%c",\*p);// t