Day 5

09 April 2024 01:02 PM

Pointer :

pointer

int *p;

int a;

p = &a;

*p = 10;

dereference

value → int

int * → address

P | &a 100

100

a | 10

Types of Variables

1. Data variable (int a)
2. Address variable (Pointer)
3. Reference variable

a = 10

cout << a; ⟶ 10

cout << p; ⟶ 100

cout << *p; ⟶ 10

```
void swap (int a, int b)
{
    int temp = a;
    a = b;
    b = temp;
}
int main()
{
    int a = 10, b = 20;
    swap(a,b);
    cout << a << b;
    return 0;
}
```

a | 10 20

b | 20 10

temp | 10

10 20

Call by value

a | 10

b | 20

Call by address :→

```
swap (int *p1, int *p2)
{
    int temp;
    temp = *p1;
    *p1 = *p2;
    *p2 = temp;
}
```

10

p1 | &a

p2 | &b

temp | 10

&a &b

call by address

10

```
    *p2 = temp;
}
int main ()
{
    int a=10, b=20;
    swap (&a, &b);
}
```

call by address

a | ~~10~~ 20 b | ~~20~~ 10

Reference Variable :→

```
int a;
add = &a
int &r = a;
```

a | ~~10~~ 20 ← r

```
cout<<a;
cout<<r;
```

```
a=10;
r=20;
```

```
int a=10;
int &r = a;
```

a | 10 ← r

```
cout<<a; → 10
cout<<r; → 10
```

&r → add

```
&r = b; ← Error
```

```
cout<<a<<r<<a<<a;
```

10 10 10 10

Call by Reference

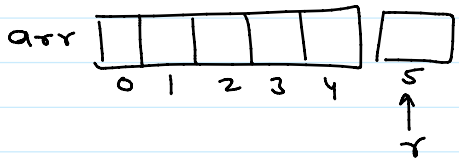
```
void swap (int &r1, int &r2)
{
    int temp;
    temp = r1;
    r1 = r2;
    r2 = temp;
}
int main ()
{
    int a=10, b=20;
```

r1 → a | ~~10~~ 20 ; temp | 10 ; r2 → b | ~~20~~ 10

```
int main()
{
    int a=10, b=20;
    swap(a,b);
```

a → 10 20 ; b → 20 10

```
int &r = a;   ✓
int &r = &a;  ✗
int arr[5];
int &r = arr; ✗
```




```
int &r = arr[5];
```

```
swap(int &, int &)
```

```
int a;
int *p1;
p1 = &a;   ✓
*p = 10;
```

a → 10 247 ; p1 → &a

```
p1 = &a;
float *p2;
p2 = &a;   ←
p2 = (float *) &a;   ✓
*p = 5.8;
cout << a;   → 0 garbage
```

Pointer to variable :-

a → 10 20 ; p1 → &a ; p2 → &p1

```
int a;
int *p1 = &a;
*p1 = 10;   → pointer to variable
int **p2 = &p1;
```

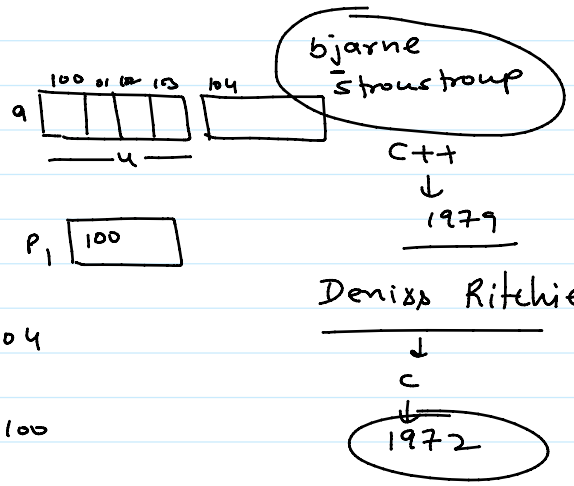
`**p2 = 20;` → pointer to pointer to variable

sizeof pointer :→

```
int *p1;        cout<< sizeof (p1);
float *p2;      cout<< sizeof (p2);
char *p3;       cout<<sizeof (p3);
double *p4;     cout<< sizeof (p4);
```

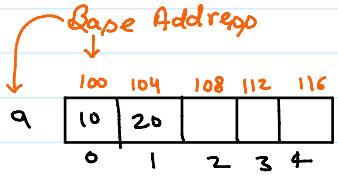
```
#include<iostream>
using namespace std;
int main()
{
    int *p1;
    float *p2;
    char *p3;
    double *p4;
    cout<<sizeof(p1)<<endl;
    cout<<sizeof(p2)<<endl;
    cout<<sizeof(p3)<<endl;
    cout<<sizeof(p4)<<endl;
    cout<<sizeof(short)<<endl;
}
```

```
int a;
int *p1 = &a;
p++;
cout<<p; → 104
p--;
cout<<p; → 100
```

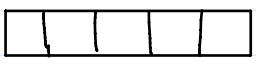


Array :→

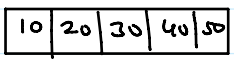
```
int a[5];
a[0] = 10;
a[1] = 20;
```




```
int a[5];   ← default Value → Garbage
```




```
int a[5] = {10, 20, 30, 40, 50};
```



```
int a[5] = {10,20};
```
a | 10 | 20 | 0 | 0 | 0 |

```
int a[5] = {0};
```
a | 0 | 0 | 0 | 0 | 0 |

```
int a[] = {10,20,30};
```
a | 10 | 20 | 30 |

```
int a[];
```
← Error

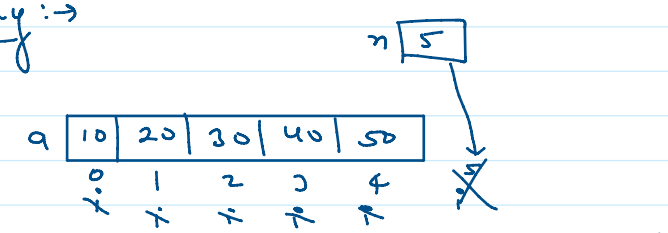
```
int a[5] = {1,2,3,4,5,6};
```
← Error

```
static int a[5];
```
a | 0 | 0 | 0 | 0 | 0 |

Input | output

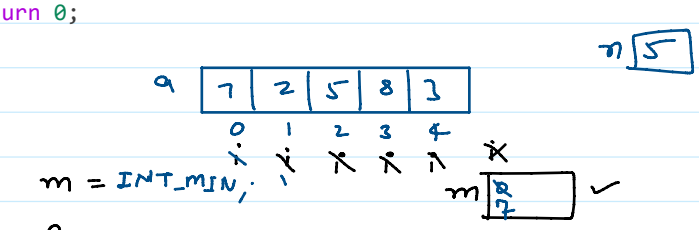
```
#include<iostream>
using namespace std;
int main()
{
    int a[10];
    int n;
    cout<<"Enter no of elements:";
    cin>>n;
    //input
    for(int i=0;i<n;i++)
    {
        cout<<"Enter value of "<<i+1<<" element:";
        cin>>a[i];
    }
    //output
    for(int i=0;i<n;i++)
    {
        cout<<a[i]<<" ";
    }

}
```

Sum of array :→

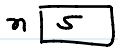



```
int sum = 0;
for (i=0; i<n; i++)
{
    sum = sum + a[i];
}
cout<<sum;
```
→ 150

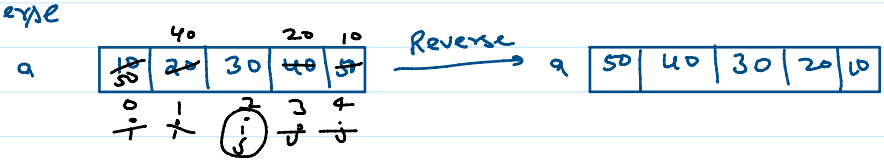
```cpp
#include<iostream>
using namespace std;
int main()
{
    int a[10];
    int n;
    cout<<"Enter no of elements:";
    cin>>n;
    //input
    for(int i=0;i<n;i++)
    {
        cout<<"Enter value of "<<i+1<<" element:";
        cin>>a[i];
    }
    //sum
    int sum=0;
    for(int i=0;i<n;i++)
    {
        sum += a[i];
    }
    cout<<"Sum of array = "<<sum;
    return 0;
}
```




```cpp
#include<iostream>
using namespace std;
int main()
{
    int a[10];
    int n;
    cout<<"Enter no of elements:";
    cin>>n;
    //input
    for(int i=0;i<n;i++)
    {
        cout<<"Enter value of "<<i+1<<"
element:";
        cin>>a[i];
    }
    //sum
    int m=INT32_MIN;
    for(int i=0;i<n;i++)
    {
        if(a[i]>m)
            m=a[i];
    }
    cout<<"Max of array = "<<m;
    return 0;
}
```



Reverse




```
for(i=0, j=n-1 ; i<j ; i++, j--)
{
```

```
for(i=0, j=n-1 ; i<j ; i++, j--)
{
        swap(a[i], a[j]);
}
```

n [5]

	50	40		20	10
a	~~10~~	~~20~~	30	~~40~~	~~50~~
	0	1	2	3	4
	i	i	i		

or

i < n-i-1

```
for(i=0; i<n/2 ; i++)
{
        swap(a[i], a[n-i-1]);
}
```

(n-i-1)

function of an array. :-

sum of array using function –

```
int sum(int *a[], int n)
{
```

Pointer

a 100

a[i] → *(a+i) : 100

```
}
```

```
int main()
{
    int a[10]
    // input
    sum(a, n);
```

n [5]

	10	20	30	40	50
a	100	104	108	112	116
	0	1	2	3	4

i = 2

a[i] = 10

*(a+i)

*(100+2)

*(108) = 10

call by Address

call by value

Array is a pointer

Pointer is an array.

✓ a[i] → *(a+i) ✓

✓ i[a] *(i+a) ✓

return 0

```
int main()
{
    cout<<"Hello";
    cout<<"India";
    return 0;
}
```

zero → Successfully

non zero → Unsuccessfully

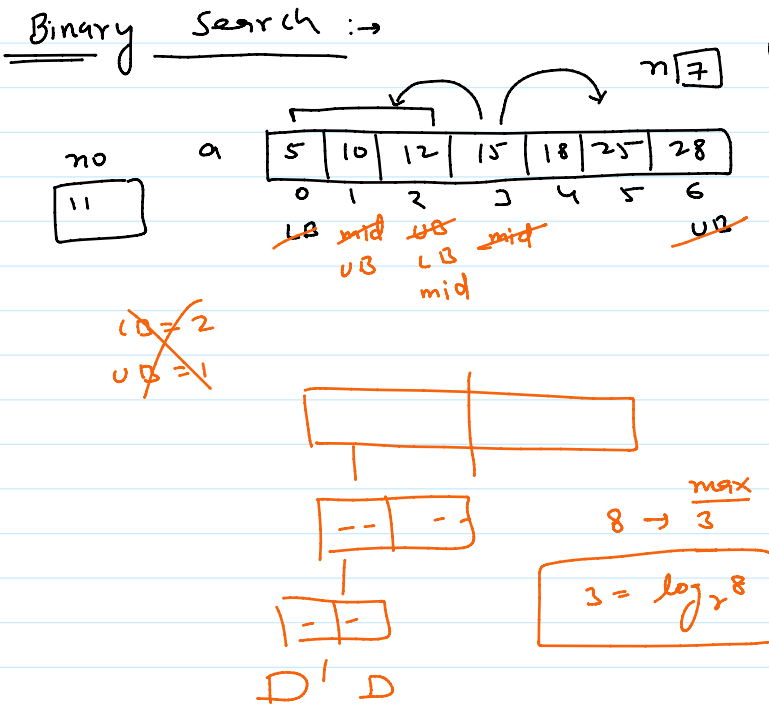
Searching :→ Linear search

n [6]

	7	5	8	3	1	2
a	0	1	2	3	4	5

```cpp
#include<iostream>
using namespace std;
bool LinearSearch(int *arr,int n,int no)
{
    for(int i=0;i<n;i++)
    {
        if(no==arr[i])
            return true;
    }
    return false;
}
int main()
{
    int a[]={5,6,8,2,1,7};
    int n=sizeof(a)/sizeof(int);
    if(LinearSearch(a,n,10))
        cout<<"Found";
    else
        cout<<"Not found";
    return 0;
}
```

Binary Search :→



Hashing

```
LB = 0
UB = n-1;
while(LB <= UB)
{
    mid = (LB+UB)/2;
    if(no == a[mid])
        return true;
    if(no < a[mid])
        UB = mid-1;
    else
        LB = mid+1;
}
return false;
```