Date 28/11/2023 Page

Day - 65

Pointers in C++ - 3

* **Pointers with char array:**

```
char arr[5] = "1234";
```

	0	1	2	3	4
arr	'1'	'2'	'3'	'4'	'\0'
	500	501	502	503	504

```
char *ptr = arr;
cout<< arr <<endl;  → 1234
cout<< ptr << endl; → 1234
```

⇒ The implementation of char is that if we pass any address then it will print the value at that address not the address

⇒ For printing the address ⟶

```
cout << (void *) arr;
cout << (void *) ptr;
```

⇒ (void *) is a pointer that points to the value but don't tell what is the data type of that value.

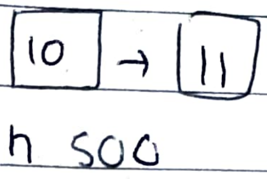
⇒ Now, if char a = 'a'; char *ptr = &a;

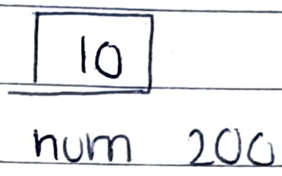
⇒ cout<< &a << endl; cout<< ptr << endl;
Then it will print the value at that address until any null character comes.

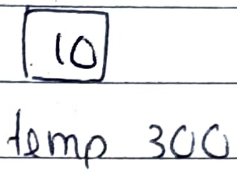
⇒ So, you can also get some random values.

Pointers in Functions

```
void incr(int n){
        n++;
}
int main(){
        int num=10;
        int temp = num;
        incr(num);
        cout<< num; → 10
}
```

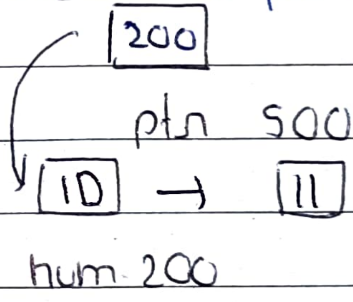






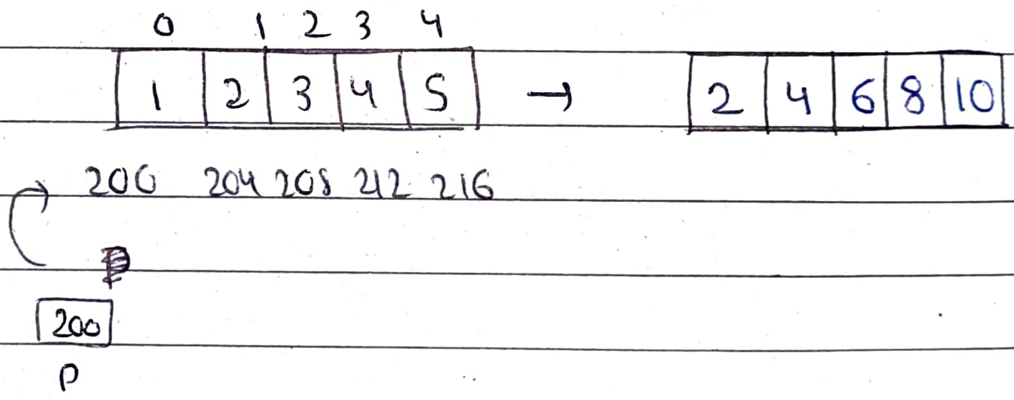
⇒ So, when we print num, we will get 10 not 11.
⇒ So for getting 11, we will have to use pointers.

```
void incr (int *ptr){
        *ptr = *ptr +1;
}
int main(){
        ⋮
        incr(&num); → 11
        ⋮
}
```



Date ______ Page ______

⇒
```
void dob( int *p){
    for( i=0; i<5; i++)
        p[i] = 2 * p[i];  →  *(p+0) =
                             2*(*(p+0))
}
int main(){
    int arr[5] = { 1, 2, 3, 4, 5 };
    dob( arr);
    for( i=0; i<5; i++)
        cout<< arr[i];
}
```



⇒ $p[1] = 2 * p[1]$

$= 2 * *(p+1)$

$*(p+1) = 2 * *(204) = 2 * 2 = 4$

Date ______ Page ______

⇒ Now, if we want to swap two no. then —

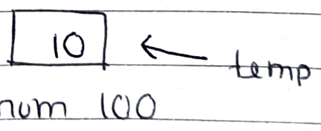
```
int main(){
    int first = 10;
    int second = 20;
    swapping (&first, &second);
    cout << first<< second ;
}

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

⇒ When we are doing operations using pointer then we have to remember many things —

⇒ So, to solve this problem, we use reference.

```
int num = 10;
int &temp = num;
temp = temp +1;
```



Here temp will also point to the num.

⇒ we can also rewrite the swapping function using reference.

```
void swapping( int &p1 , int &p2){
        int temp = p1;
        p1 = p2;
        p2 = temp ;
}
```