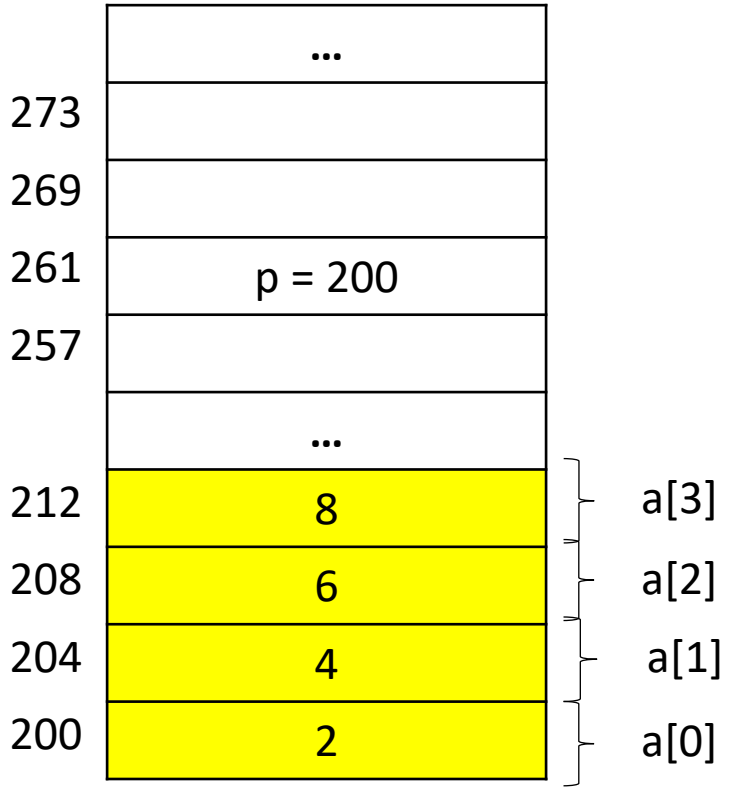
2D arrays and Pointers

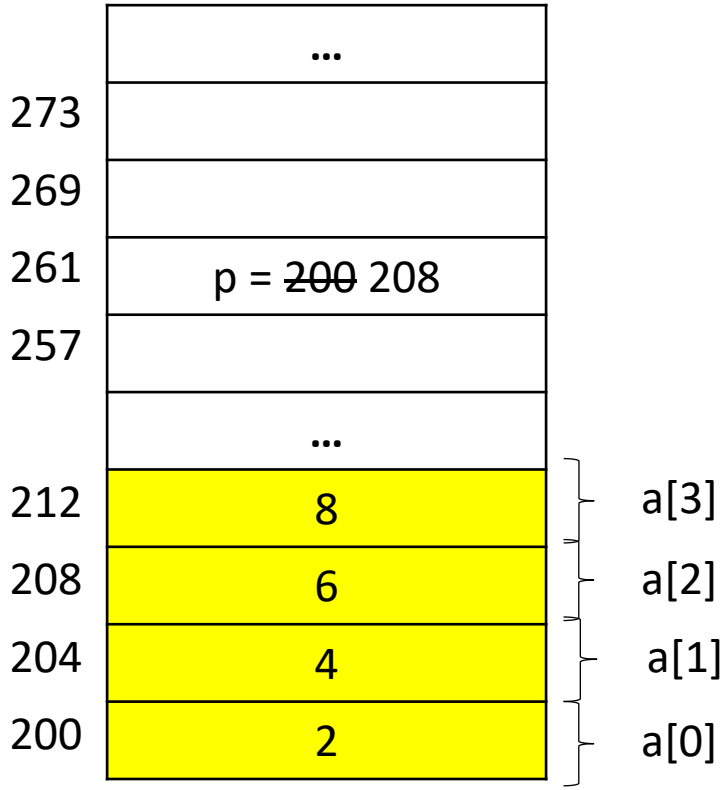
Pointers and 1D Arrays

```
int a[] = {2, 4, 6, 8};

int *p = a; //Equivalent to *p = &a[0]

printf("%d , %d", (p+1), *(p+1)); //204, 4
printf("%d , %d", (a+1), *(a+1)); //204, 4
printf("%d , %d", &a[1], a[1]); //204, 4
```



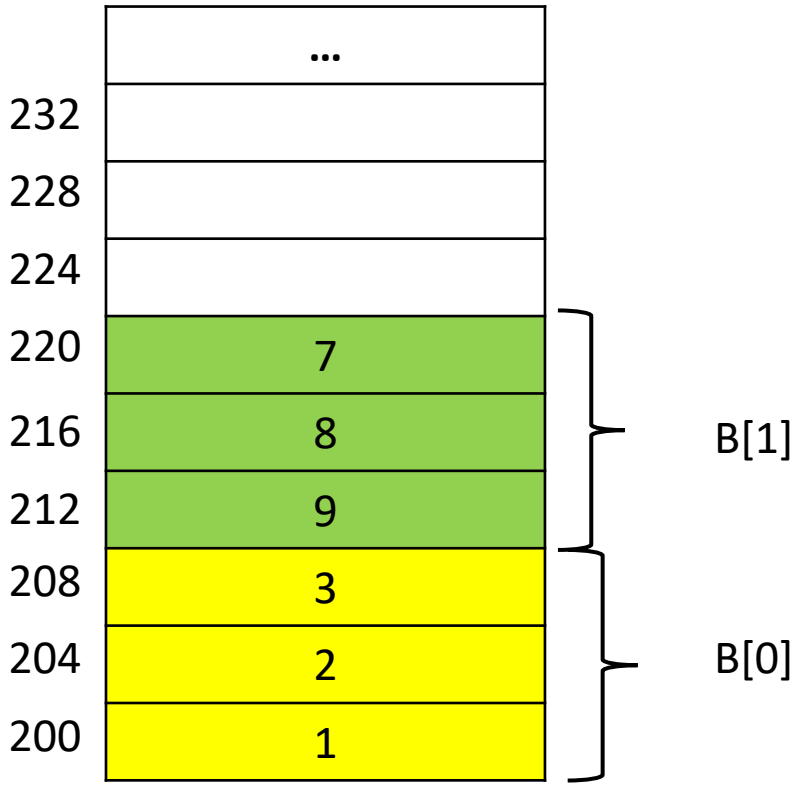
Pointers and 1D Arrays

```
int a[] = {2, 4, 6, 8};

int *p = a; //Equivalent to *p = &a[0]

printf("%d , %d", (p+1), *(p+1)); //204, 4
printf("%d , %d", (a+1), *(a+1)); //204, 4

p = p + 2
printf("%d , %d", p, *p); //208, 6

a = a + 2; //INVALID. a always points to the
base address
```



Pointers and 2D Arrays

```
int B[2][3] = {{1,2,3},{9,8,7}};
//B[0] and B[1] are 1D arrays of 3 integers

int *p = B;
```



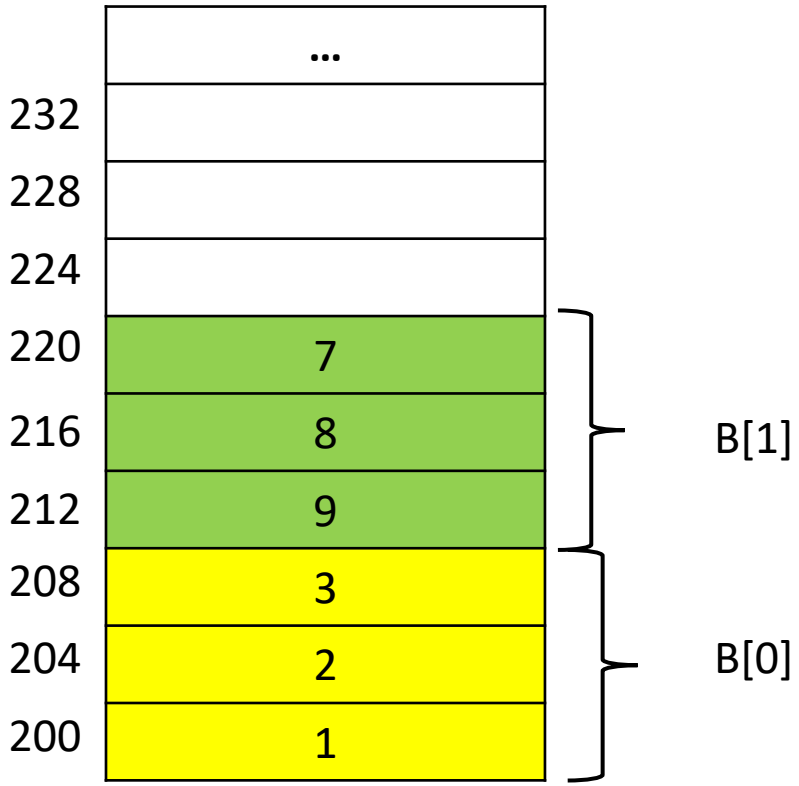
Pointers and 2D Arrays

```
int B[2][3] = {{1,2,3},{9,8,7}};
```

//B[0] and B[1] are 1D arrays of 3 integers

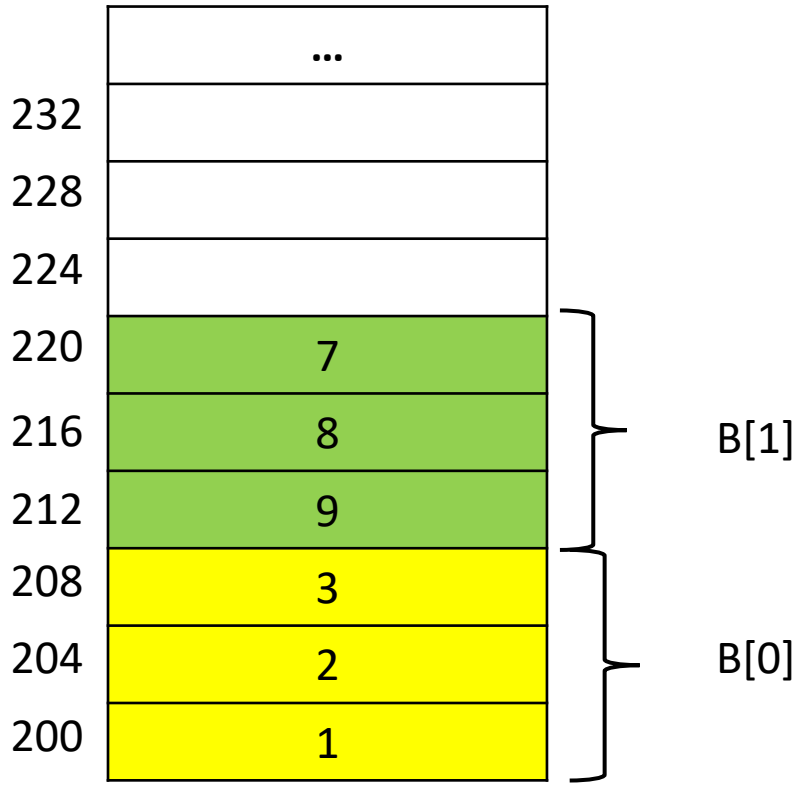
```
int *p = B; //Compilation ERROR. B is a pointer
to 1D array of 3 integers
```

```
int *p[3] = B; //Correct
```



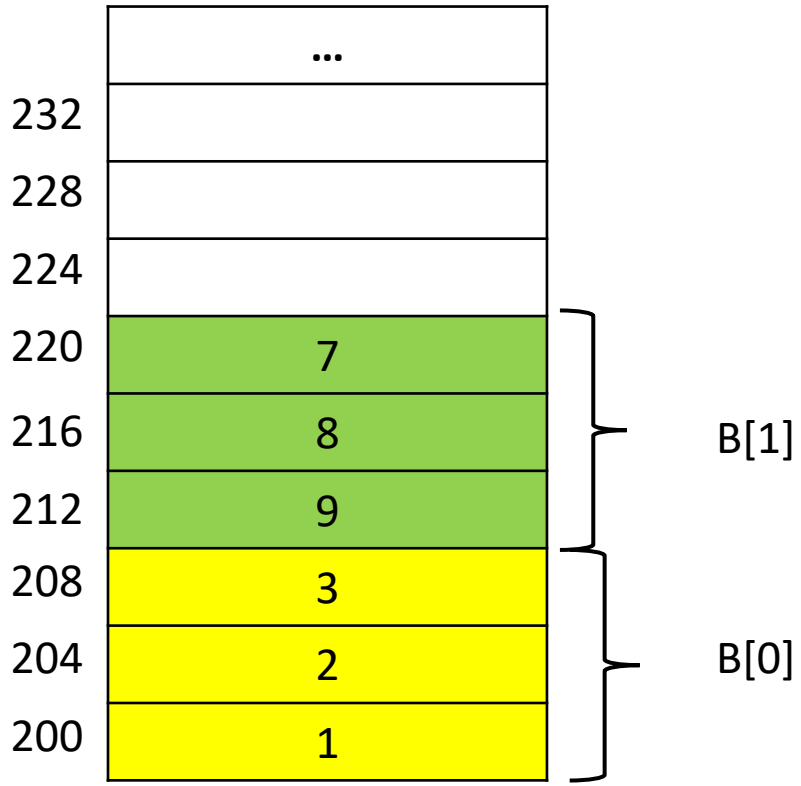
Pointers and 2D Arrays

```
int B[2][3] = {{1,2,3},{9,8,7}};
//B[0] and B[1] are 1D arrays of 3 integers

int *p = B; //Compilation error. B is a pointer to
1D array of 3 integers
int *p[3] = B; //Correct

printf("%d\n", B); //&B[0] = 200
printf("%d\n", *B); //*(&B[0])= B[0]= &B[0][0]= 200
```



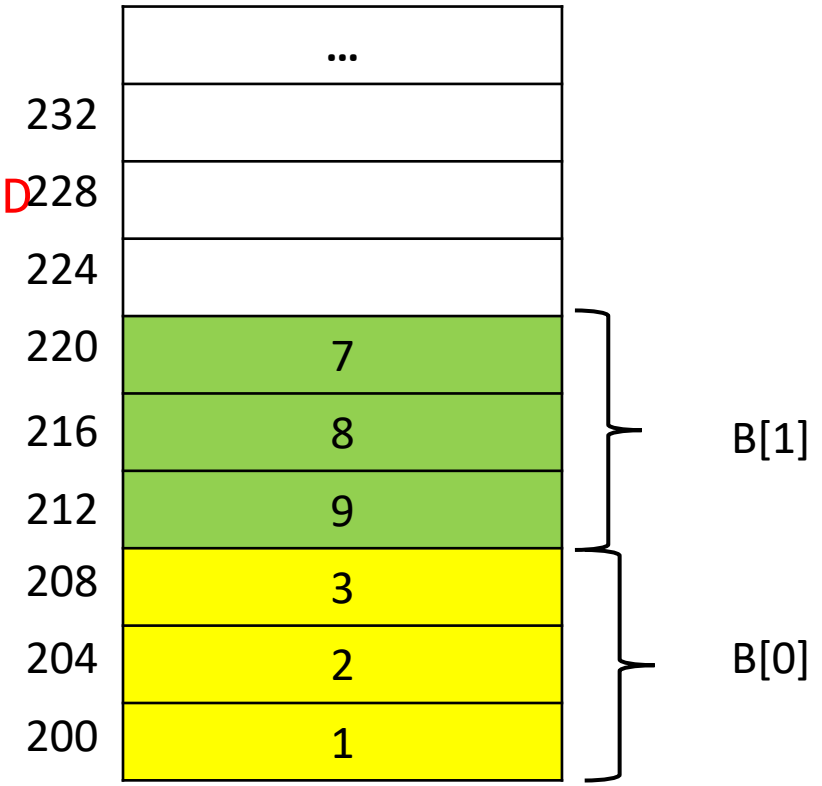
Pointers and 2D Arrays

```
int B[2][3] = {{1,2,3},{9,8,7}};
//B[0] and B[1] are 1D arrays of 3 integers

int *p = B; //Compilation error. B is a pointer
to 1D array of 3 integers
int *p[3] = B; //Correct

printf("%d\n", B); //200 or &B[0]
printf("%d\n", *B); //B[0] or &B[0][0] or 200
```



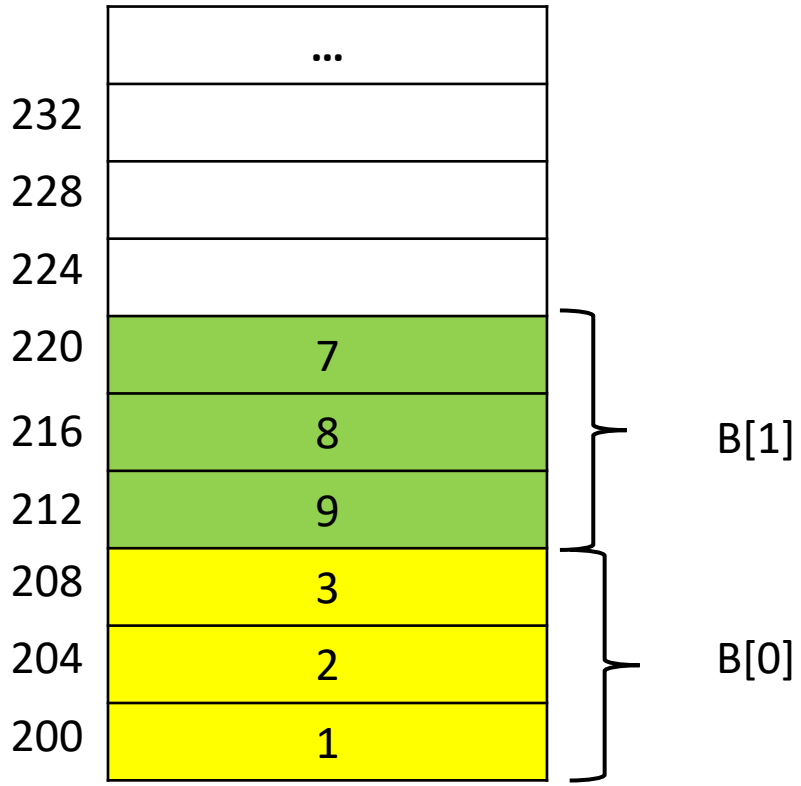
Pointers and 2D Arrays

```
int B[2][3] = {{1,2,3},{9,8,7}};
//B[0] and B[1] are 1D arrays of 3 integers

int *p = B; //Compilation error. B is a pointer to 1D array of 3 integers
int *p[3] = B; //Correct

printf("%d\n", B); //200 or &B[0]
printf("%d\n", *B); //B[0] or &B[0][0] or 200

printf("%d\n", B + 1); //&B[0]+sizeof(B[0])= 200+12
printf(%d\n", *(B+1) + 2));
//*(B+1)+2 = *(&B[1])+2 = &B[1][0]+2
= 212+2*sizeof(int) = 212+8 = 220
```



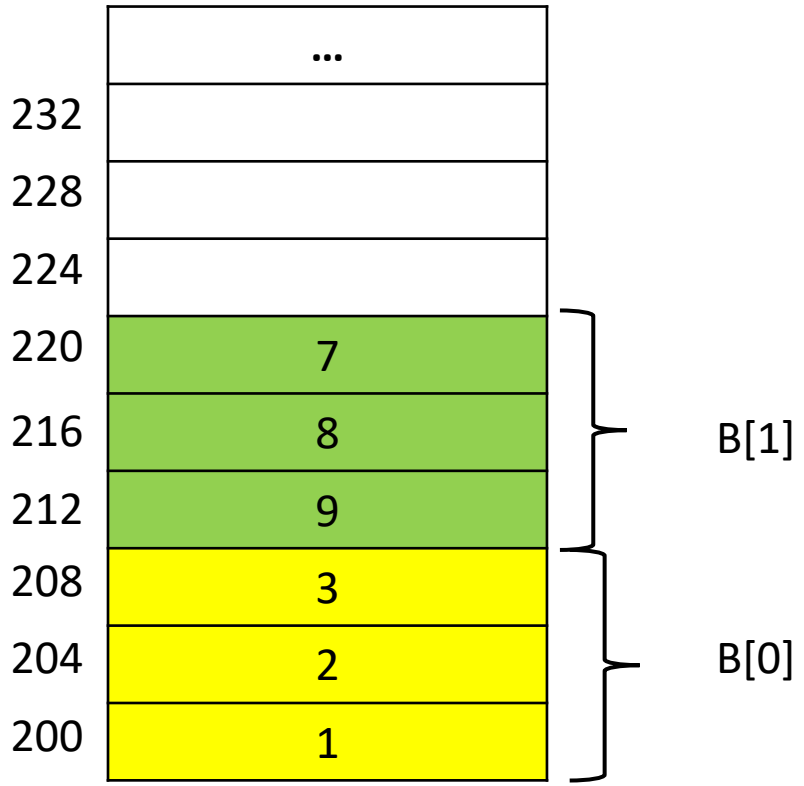
Pointers and 2D Arrays

```
int B[2][3] = {{1,2,3},{9,8,7}};
//B[0] and B[1] are 1D arrays of 3 integers

int *p = B; //Compilation error. B is a pointer
to 1D array of 3 integers
int *p[3] = B; //Correct

printf("%d\n", B); //200 or &B[0]
printf("%d\n", *B); //B[0] or &B[0][0] or 200

printf("%d\n", B + 1); //212 or &B[1]
printf(%d\n", *(B+1) + 2);
//&B[1][0]+2*4 = &B[1][2] = 220
```



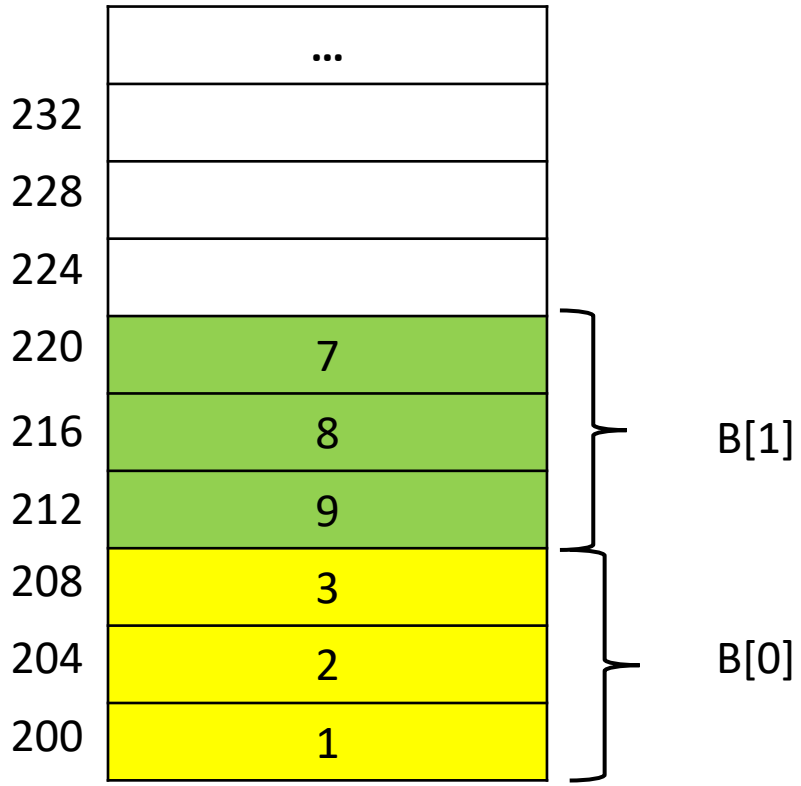
Pointers and 2D Arrays

```
int B[2][3] = {{1,2,3},{9,8,7}};
//B[0] and B[1] are 1D arrays of 3 integers

int *p = B; //Compilation error. B is a pointer
to 1D array of 3 integers
int *p[3] = B; //Correct

printf("%d\n", B); //200 or &B[0]
printf("%d\n", *B); //B[0] or &B[0][0] or 200
printf("%d\n", B + 1); //212 or &B[1]
printf(%d\n", *(B+1) + 2); //&B[1][0]+2 = 220

printf(%d\n", *(*B+1));
//*(*B+1)=*(*(&B[0])+1)=
*(&B[0][0]+1*sizeof(int)) = *(&B[0][1])= B[0][1]
```



Pointers and 2D Arrays

```
int B[2][3] = {{1,2,3},{9,8,7}};
//B[0] and B[1] are 1D arrays of 3 integers

int *p = B; //Compilation error. B is a pointer
to 1D array of 3 integers
int *p[3] = B; //Correct

printf("%d\n", B); //200 or &B[0]
printf("%d\n", *B); //B[0] or &B[0][0] or 200
printf("%d\n", B + 1); //212 or &B[1]
printf(%d\n", *(B+1) + 2); //&B[1][0]+2 = 220

printf(%d\n", *(*B+1));
//*(&B[0][0]+1) = *(&B[0][1]) = 2
```



Pointers and 2D Arrays

$$A[i][j] = *(A[i] + j) = *(*(A+i) + j)$$

References

- Pointers in C/C++ - YouTube