→ Stack and heap

→ loops

→ Dashboard Question

Memory management

RAM

```
b s  void main( ... ) {

    int a;
      a = 2;
    int b = 3;
    int c = b;
    int arr[] new int[5];
    arr[2] = 10;
    int arr2 [];
      arr2 = arr;
}
```

① Function calls

② storing primitive data type

loops

```
for (int i = 1;  i <= 10;  i++)
        Syso(i * n);
```

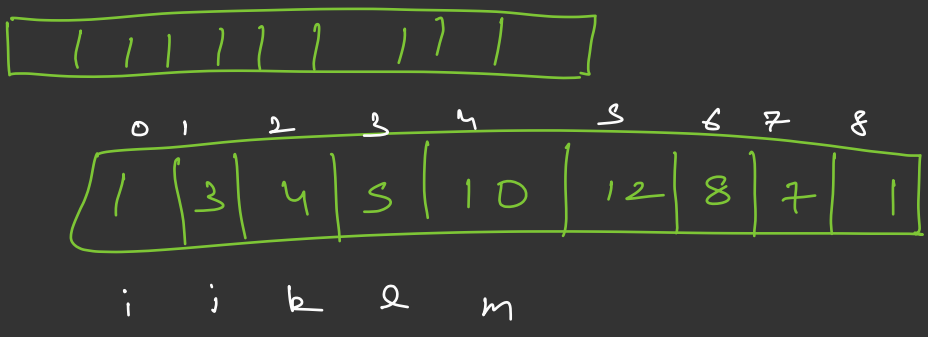
$$\frac{a + b + c + d \ldots\ldots n}{n}$$

n

1. Take input
2. Add

n x 1
n x 2
n x 3
n x 4
n x 5
n x 6
:
n x 10

Given an array of N integers find the number of mountains in the array. A mountain is a tuple (i,j,k,l,m) such that i < j < k < l < m and Arr[i] < Arr[j] < Arr[k] > Arr[l] > Arr[m].
As the number of such mountains can be large print number of mountains modulo 1000000007




```
for(int i=0; i<a.length-4; i++){
  for(int j=i+1; j<a.length-3; j++){
    for(int k=j+1; k<a.length-2; k++){
      for(int l=k+1; l<a.length-1; l++){
        for(int m=l+1; m<a.length; m++){
          if(arr[i]<arr[j] && arr[j]<arr[k]
             && arr[k]>arr[l] &&
             arr[l]>arr[m]){
            count++;
  }}}}}
```

```
for (int i=1; i<=5; i++) {
    if (i==3) {
        continue;
    }
    syso (i);
}
```

i = 1 2 ~~3~~

1 2

int → 4 bytes ⇒ 32 bits

2^{32}

-2^{31} to $2^{31}-1$

0 / 1

$2^{31} = 2^{10} \times 2^{10} \times 2^{10} \times 2$

$2^{10} = 1024$

$2^{10} \approx 1000$

10^3

$\approx 10^3 \times 10^3 \times 10^3 \times 2$

$\approx 10^9 \approx 1000000007$

2109 %. 10 $\Rightarrow$ 9

9 %. 10 = 9

%. 2 $\Rightarrow$ 0, 1

```
f = 0
sl = 0
tl = 0
```

1 5 2 4 9 3 10

```
if( f < arr[i]) {
    tl = sl;
    sl = f;
    f = arr[i];

} else if (sl < arr[i])
{   tl = sl;
    sl = arr[i];

} else if (tl < arr[i]) {
    tl = arr[i];
} }
```