When you pass array in function you are passing reference so in function definition another variable get created but it is pointing to original data because reference get copied.[Arrow get copied]

Lets solve this question accept data in an array pass it to function and function should store square of each element in the same array

Eg. Input 5 3 2

o/p 25 9 4 this value get replace in original array

Stack

|  |  |  |
| --- | --- | --- |
| 5 25 | 3 9 | 2 4 |

Heap

2000

arr

**import** **static** java.lang.System.***out***;

**import** java.util.\*;

2000

**public** **class** Myclass **{**

**public** **static** **void** main(String[] args) {

Scanner sc=**new** Scanner(System.***in***);

**int**[] arr;

Stack

***out***.print("enter size of array");

a

**int** size=sc.nextInt();//3

arr=**new** **int**[size];

2000

**int** i;

**for**(i=0;i<arr.length;i++)

a

{

Int a=5;

Int b;

b=a;

arr[i]=sc.nextInt();//5 3 2

5

}

*disparr*(arr);

**for**(**int** n:arr)

b

***out***.println(n);

}

5

**public** **static** **void** disparr(**int** [] a)

{ **for**(**int** i=0;i<a.length;i++)

a[i]=a[i]\*a[i];

}

**}**

Accept 5 element in an array. // 1 3 2 4 5

2. call method factofarr(int [] a) which will store factorial of each element in same array 1 6 2 24 120

3.call method printarr(int [] a) which will print all element// 1 6 2 24 120

4. call method sumofarr(int [] a) which will do sum of each element[] //153

Q🡺 Accept 5 element in an array and copy this data in to another array.

Stack

|  |  |  |
| --- | --- | --- |
| 5 | 3 | 2 |

Heap

2000

arr

**import** **static** java.lang.System.***out***;

**import** java.util.\*;

**public** **class** Myclass {

**public** **static** **void** main(String[] args) {

Scanner sc=**new** Scanner(System.***in***);

**int**[] arr,cpy;

cpy=arr;

this will lead to shallow copy

***out***.print("enter size of array");

**int** size=sc.nextInt();//3

arr=**new** **int**[size];

cpy=**new** **int**[size];

Stack

|  |  |  |
| --- | --- | --- |
| 5 | 3 | 2 |

Heap

4000

cpy

**int** i;

**for**(i=0;i<arr.length;i++)

{

arr[i]=sc.nextInt();//5 3 2

cpy[i]=arr[i];

}

**for**(**int** n:cpy)

***out***.println(n);

}

Shallow copy

**import** **static** java.lang.System.***out***;

**import** java.util.\*;

**public** **class** Myclass {

**public** **static** **void** main(String[] args) {

Stack

|  |  |  |
| --- | --- | --- |
| 5 | 3 | 2 |

Heap

2000

arr

Scanner sc=**new** Scanner(System.***in***);

**int**[] arr,cpy;

***out***.print("enter size of array");

**int** size=sc.nextInt();//3

arr=**new** **int**[size];

cpy=**new** **int**[5];

**int** i;

Stack

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 |

Heap

2000

cpy

**for**(i=0;i<arr.length;i++)

{ arr[i]=sc.nextInt();//5 3 2

}

**for**(**int** n:cpy)

***out***.println(n);

cpy=arr;

System.out.println(cpy.hashCode());

**for**(**int** n:cpy)

***out***.println(n); //532

}

Q Accept 5 element in an array and store even element in to another array

Stack

|  |  |  |
| --- | --- | --- |
| 4 | 3 | 2 |

Heap

2000

arr

**import** **static** java.lang.System.***out***;

**import** java.util.\*;

**public** **class** Myclass {

**public** **static** **void** main(String[] args) {

Scanner sc=**new** Scanner(System.***in***);

**int**[] arr,cpy;

cpy=arr;

this will lead to shallow copy

***out***.print("enter size of array");

**int** size=sc.nextInt();//3

arr=**new** **int**[size];

cpy=**new** **int**[size];

Stack

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 2 |
| 4 | 2 | 0 |

Heap

4000

cpy

**int** i,j;

**for**(i=0,j=0;i<arr.length;i++)

{

arr[i]=sc.nextInt();//4 3 2

if(arr[i]%2==0)

cpy[j++]=arr[i];

}

**for**(i=0;i<j;i++)

***out***.println(cpy[i]);//4 2

}

Q🡺 Accept 5 element in an array and store square of each element in to another array

**import** **static** java.lang.System.***out***;

**import** java.util.\*;

**public** **class** Myclass {

**public** **static** **void** main(String[] args) {

Scanner sc=**new** Scanner(System.***in***);

**int**[] arr,cpy;

***out***.print("enter size of array");

**int** size=sc.nextInt();//3

arr=**new** **int**[size];

cpy=**new** **int**[size];

Stack

|  |  |  |
| --- | --- | --- |
| 5 | 3 | 2 |

Heap

2000

arr

**int** i;

**for**(i=0;i<arr.length;i++)

{

arr[i]=sc.nextInt();//5 3 2

cpy[i]= arr[i]\*arr[i];

}

**for**(**int** n:cpy)

***out***.println(n);

System.out.println(cpy.hashCode());

Stack

|  |  |  |
| --- | --- | --- |
| 25 | 9 | 4 |

Heap

4000

cpy

**for**(**int** n:cpy)

***out***.println(n);

}

Q🡺 Accept 5 element in an array and do sum of it . Also print average of all the element.

Class Main(){

public static void main(String[] args) { Scanner sc=new Scanner(System.in); int a[]= new int[5]; for (int i=0;i<=4;i++){ a[i]=sc.nextInt(); } int sum=0; for (int i=0;i<=4;i++){ sum = sum+a[i]; } System.out.println(sum);

float avg = (float)sum/a.length;

System.out.println(avg);

}

Q🡺Accept 5 element in an array and print highest element

Q🡺Accept 5 element in array and accept start index accept number of element to be extracted from the array and store the extracted element in to another array.

5 6 8 9 2🡺 stat index 🡺2 number of element extract🡺3 O/P🡺8 9 2

Can function / method return an array?

Yes.

Accept a number pass it to method let method return all factor of that number as array.

**import** **static** java.lang.System.***out***;

**import** java.util.\*;

no

**public** **class** Myclass {

**public** **static** **void** main(String[] args) {

12

Scanner sc=**new** Scanner(System.***in***);

**int**[] arr;

***out***.print("enter a number ");

**int** no=sc.nextInt();//12

arr

arr=*getfact*(no);

a

**for**(**int** i=0;i<arr.length ;i++)

12

***out***.println(arr[i]);

}

arr

**public** **static** **int**[] getfact(**int** a)//12

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | **1** | **2** | **3** | **4** | **5** | **6** |
| 1 | 2 | 3 | 4 | 6 | 12 | 0 |

{

**int** []arr=**new** **int**[a/2+1];

**int** j=0;

stack

**for**(**int** i=1;i<=a;i++)

{

Heap

**if**(a%i==0)//12%1 12%2

arr[j++]=i;//6

}

**return** arr;

}

}