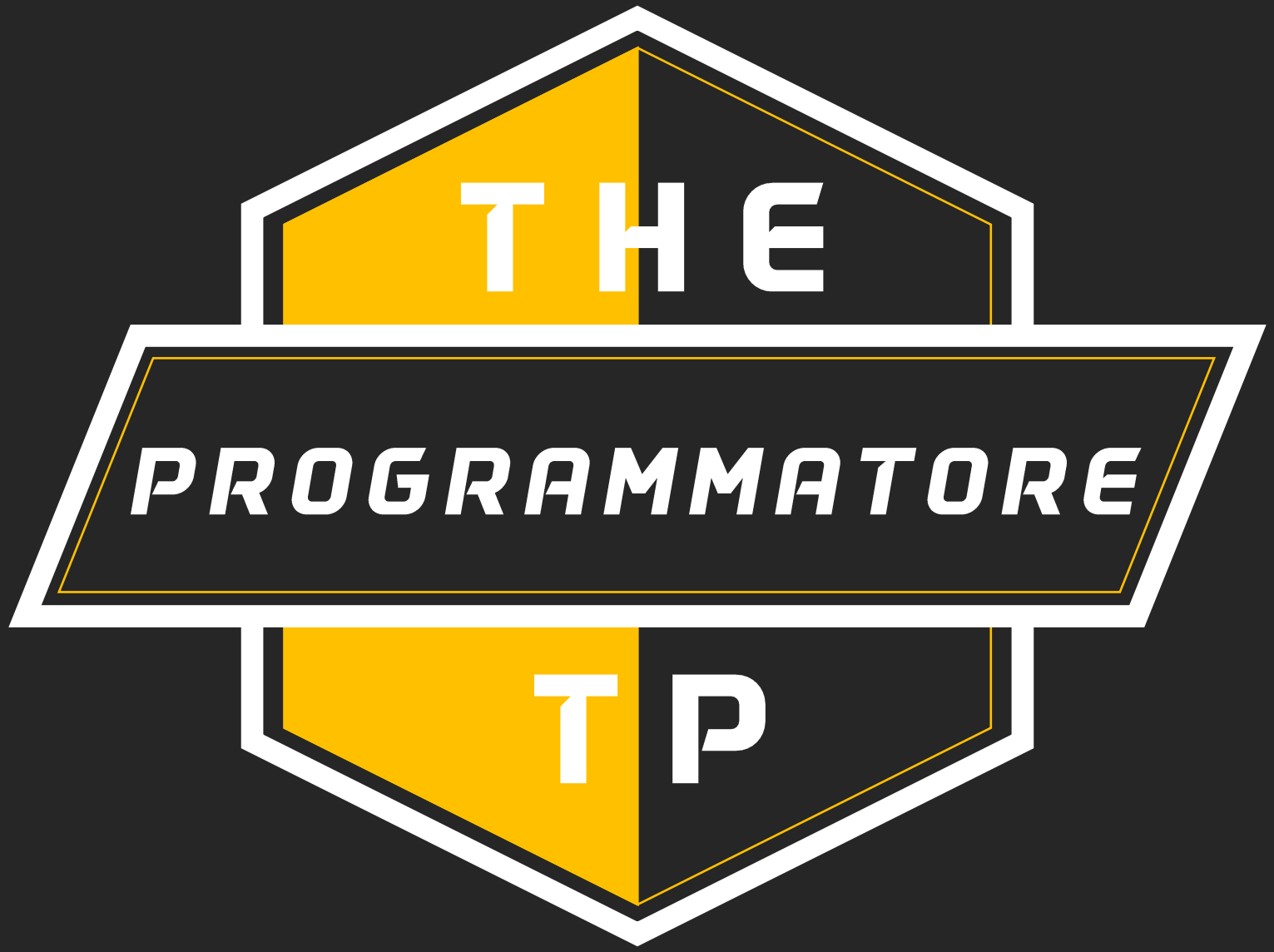


“A small step in the world of Programming”

Outlines

- *Definition of array*
- *Understanding the definition of array*
- *Introduction to one dimensional array*

Array

- An array is a *data structure* that contains a group of elements.
- Typically these elements are all of the same ***data type***, such as an ***integer***, ***float/double*** or ***string***.

Example

A search engine may use an array to store Web pages found in a search performed by the user. When displaying the results, the program will output one element of the array at a time.

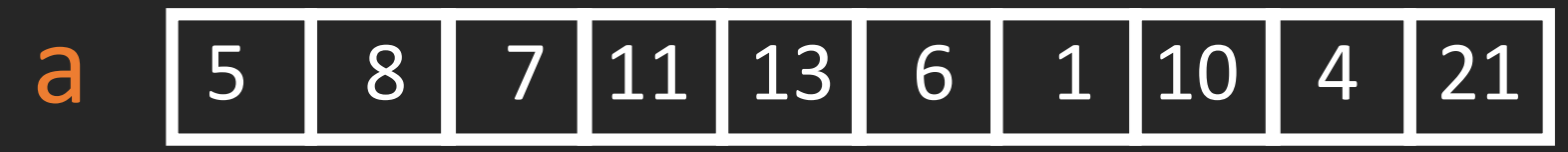
Storing the results in an array is much more efficient way to manage memory.

- take 10 variables of integer types

```
int  a, b, c, d, e, f, g, h, i, j
```

- What if we can take a better approach?

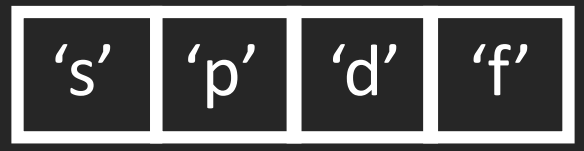
```
int a[10]
```



Mark the correct array

a

a



a

Array

One Dimensional Array

```
data_type array_name[number_of_variable]
```

```
int arr[5]
```

*Compiler will allocate a contiguous block of memory size = 5 * sizeof(int)*

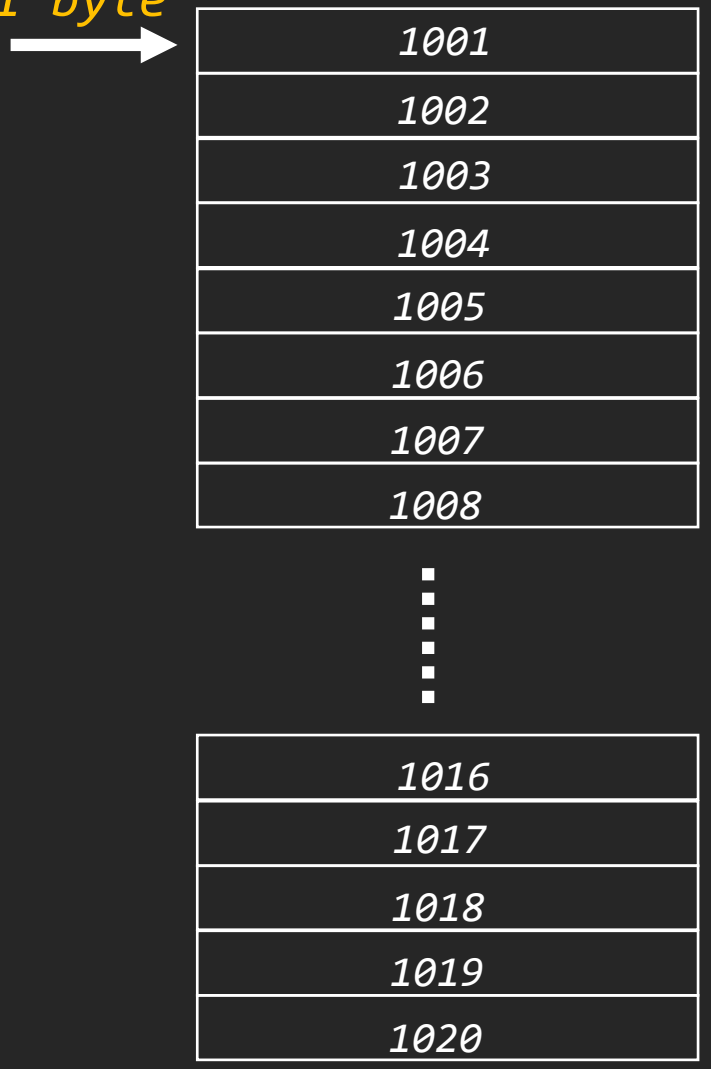
Array


```
arr[0] = 5
arr[1] = 7
arr[2] = 11
arr[3] = 2
arr[4] = 4
```

Array


```
int arr[5]
```

each block = 1 byte



int = 4 bytes

arr[0] = 1001 - 1004

arr[1] = 1005 - 1008

arr[2] = 1009 - 1012

arr[3] = 1013 - 1016

arr[4] = 1017 - 1020

Mark the correct array declaration

```
int a[10]
```

```
int a[5+3]
```

```
int a[6/2]
```

```
int a[3/2]
```

```
int a[3*2]
```

```
int a[b=(6%2)+2]
```

```
int a[-4]
```

❑ Assigning Value into an Array

- *Assigning value while declaring array*
- *Assigning value after declaring array*
- *Assigning value in runtime*

Array

- *Assigning value while declaring array*

```
int arr[5] = {3, 5, 7, 9, 10};

int arr[] = {3, 5, 7, 9, 10};
```

Array

- Assigning value after declaring array

```
int arr[5];

arr[0] = 3
arr[1] = 5
arr[2] = 7
arr[3] = 9
arr[4] = 10
```

Array

- *Assigning value in runtime*

```
int arr[5];

for(i=0; i<5; i++){
     scanf("%d", &arr[i]);
}
```

```
i=0, condition true
arr[0] -> 3

i=1, condition true
arr[1] -> 5

i=2, condition true
arr[2] -> 7

i=3, condition true
arr[3] -> 8

i=4, condition true
arr[4] -> 10

i=5, condition false
exit from loop
```

Array

- *Printing the array*

```
for(i=0; i<5; i++){
        printf("%d\n", arr[i]);
}
```

What if number of elements are lesser than the length specified?

```
int arr[10] = {3, 5, 7, 9, 10};
```

Answer

The remaining locations of the array are filled with value 0.

```
int arr[10] = {3, 5, 7, 9, 10, 0, 0, 0, 0, 0};
```

Another good way to determine array size is using #define

```
#define Size 20

int arr[Size];
```

```
To initialize all the value of array 0.

int arr[10] = {0};
```

Examples

- *A program that print an array in reverse order.*
- *A program to calculate sum of all numbers in the array.*
- *A program to find a specific number from an array.*
- *A program to find the number of even numbers in an array.*
- *A program to copy all the elements into another array.*
- *A program to find number of vowels in an array.*
- *A program to print results of all students with roll number.*