Session 1

ACM - ASUFE CPC

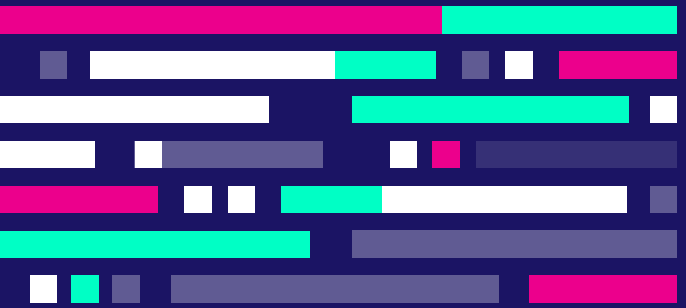
Introduction to CPP Programming & Problem Solving

Session Content

1. Introduction: What's problem solving?
2. What's CPP? What's an IDE?
3. First program: Adding two numbers
4. Data types & arithmetic operators
5. Control flow:
    a) If and else statements, switch case, relational operators
    b) While and for loops, continue and break
6. Arrays & strings

Session Content

1. What's problem solving?



Problem:

Given two numbers (x and y), you need to find their sum.

What's the input? The two numbers: x & y

What's the output? Their sum

How could you solve this?

Sum = a + b, done on a calculator

2. What's CPP?

CPP is a programming language, and any programming language is a tool the programmer uses to communicate with the computer in order to solve a specific problem.

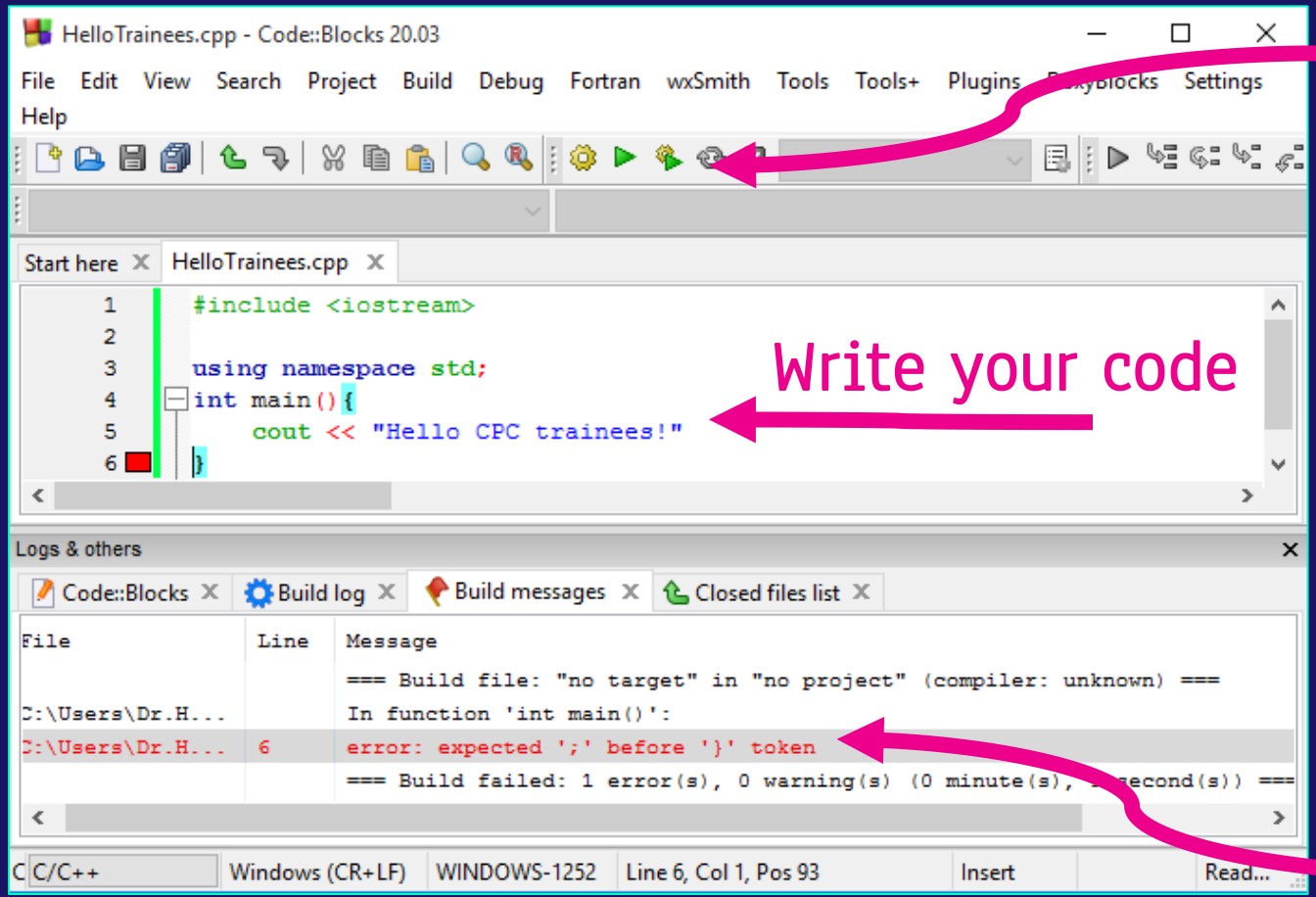
2. What's an IDE?

An IDE -Integrated Development Environment- is another tool a programmer uses to make development much easier.

Some popular IDEs are: CodeBlocks, Visual Studio, Eclipse, Atom, and there are many, many more.

Check these out! [1] [2] [3]

2. What's an IDE?



Build and run your code

Write your code

Fix errors if any

On namespaces.

2. What's an IDE?

All IDEs are somewhat similar in how they look.

Until you setup your IDE, you might use an online one:
Programiz, OnlineGDB, C++ Shell, and there are many more out there as well.

3. FIRST PROGRAM: ADDING X & Y

```cpp
#include <iostream> //This is the library we need to use cout and cin

using namespace std;

int main(){
    int x, y; //Defining each variable, semicolon after each line
    cin >> x >> y; //Taking two inputs and putting them in x & y
    int sum; //Defining the variable we'll store the sum in
    sum = x + y; //Calculating the sum
    cout << sum; //Printing our output

    return 0;
}
```

```
4 6
10
Process returned 0 (0x0)   execution time : 0.889 s
Press any key to continue.
```

Problem:

Zaghloul was daydreaming about his salary in 2030.

Given his salary now as a fresh grad, print his salary in 2030, knowing that it would be 5 times his current salary.

```cpp
#include <iostream>

using namespace std;

int main(){
    int zaghloulSalary;
    cin >> zaghloulSalary;
    cout << zaghloulSalary * 5;
    return 0;
}
```

```
5000
25000
Process returned 0 (0x0)
Press any key to continue.
```

04

Data Types & Arithmetic Operators

4. Data Types

Data type	Use	Examples	Limits
int	For integers (whole numbers)	1000	-(2^32) to (2^32)-1 $\pm 2 \times 10^9$
long long	For bigger integers	1,000,000,000,000,000	-(2^63) to (2^63)-1 $\pm 9 \times 10^{18}$
float	For floating point numbers. Don't use it.	2.5	-3.4×10^{38} to $+3.4 \times 10^{38}$
double	For floating point numbers.	2.5	-1.7×10^{308} to $+1.7 \times 10^{308}$
char	For single characters.	'm'	
bool	Boolean value	1 or 0	

Cool resource.

ASCII Table

33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	|
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	▯
48	0	64	@	80	P	96	`	112	p	128	€

Problem:

Zaghloul is having a hard time remembering the order of the alphabet.

Given a letter, print the letter after it in alphabetical order.

Note that this solution will fail when the letter is z.

```cpp
#include <iostream>

using namespace std;

int main(){
    char letter;
    cin >> letter;
    letter = letter + 1;
    cout << letter;
    return 0;
}
```

```
a
b
Process returned 0 (0x0)
Press any key to continue.
```

Good resource.

4. Arithmetic & assignment operators

Operator	Operation
`+`	Addition
`-`	Subtraction
`*`	Multiplication
`/`	Division
`%`	Modulo Operation (Remainder after division)

`*=`	`a *= b;`	`a = a * b;`

Operator Precedence

( ) [ ]	Operators within parenthesis are performed first	Higher
++, --	Postfix increment / decrement	
++, --	Prefix increment / decrement	
*, /, %	Multiplication, Division, Modulus	
+, -	Addition, Subtraction	
<, <=, >, >=	Less than, Less than or equal to, Greater than, Greater than or equal to	
==, !=	Equal to, Not equal to	
&&	Logical AND	
||	Logical OR	
?:	Conditional Operator	
=	Simple Assignment	
+=, -=, *=, /=	Shorthand operators	
,	Comma operator	Lower

More on that here.

Hands-on practice time!

Problems discussed by mentors were:

1- Banknotes:
https://www.beecrowd.com.br/judge/en/problems/view/1018

2- Weekdays:

Every day of the week is denoted by a number, 0 for Saturday, 1 for Sunday, 2 for Monday, etc. till Friday, number 6.

If today is Thursday, what day will it be after X days?

Input: A single integer number **x**, $0 \leq x \leq 1000$.
Output: A single integer number, the day's code.

Solutions in next slide. Please try your best before moving on to the solution.

Problems discussed by mentors were:

1- Banknotes:

```cpp
#include <iostream>
using namespace std;

//Banknotes
int main(){
    int N;
    cin >> N;
    cout << N / 100 << " nota(s) de R$ 100,00" << endl;
    N = N % 100;
    cout << N / 50 << " nota(s) de R$ 50,00" << endl;
    N = N % 50;
    cout << N / 20 << " nota(s) de R$ 20,00" << endl;
    N = N % 20;
    cout << N / 10 << " nota(s) de R$ 10,00" << endl;
    N = N % 10;
    cout << N / 5 << " nota(s) de R$ 5,00" << endl;
    N = N % 5;
    cout << N / 2 << " nota(s) de R$ 2,00" << endl;
    N = N % 2;
    cout << N << " nota(s) de R$ 1,00" << endl;
    return 0;
}
```

Problems discussed by mentors were:

2- Weekdays:

```cpp
#include <iostream>

using namespace std;

//Weekday
    /* Today's Thursday, so today is 5.
    So after 1 day, output is 6.
    After 2 days, output is 0.
    After 3 days, output is 1.
    Notice the pattern? */

int main(){
    int x;
    cin >> x;
    cout << (5 + x) % 7;
    return 0;
}
```

Careful with integer/double operations!

If both operands are integers, the result will always be an integer, no matter where you store it.

For example:

```
int x = 5, y = 2;
double z = x / y;
```

So, use casting!

```
int x = 5, y = 2;
double z = (double) x / y;
```

Variable naming

The general rules for naming variables are:

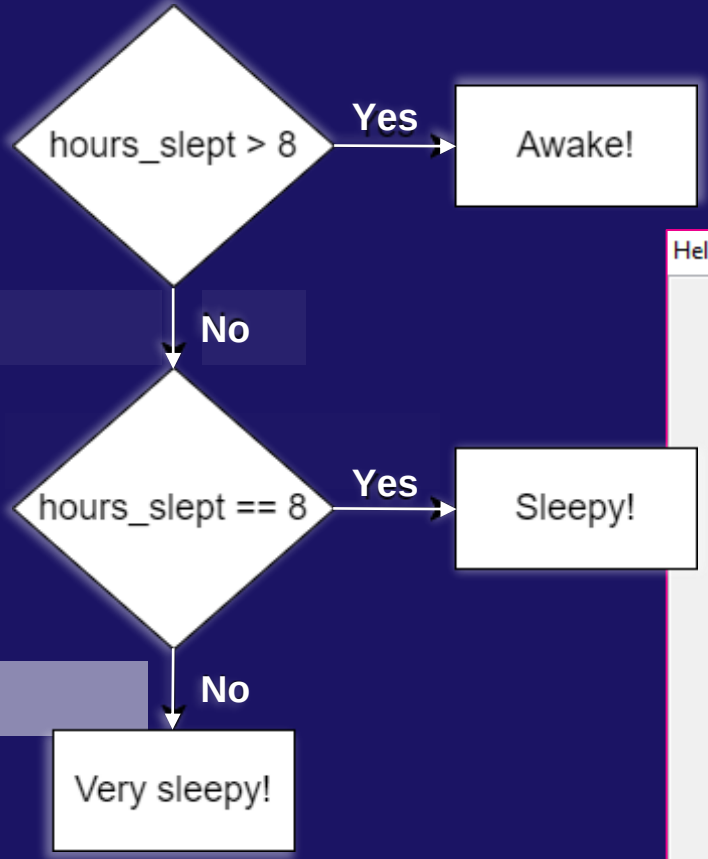
- Names can contain letters, digits and underscores
- Names must begin with a letter or an underscore (_)
- Names are case sensitive (`myVar` and `myvar` are different variables)
- Names cannot contain whitespaces or special characters like !, #, %, etc.
- Reserved words (like C++ keywords, such as `int`) cannot be used as names

Taken from here.

05. A) Control flow: New problem

Zaghloul will NOT be sleepy during his lectures tomorrow if he sleeps more than 8 hours, else, if he sleeps exactly 8 hours he'll be slightly sleepy, otherwise he'll be very sleepy.

Given the number of hours he'll sleep, print "Awake!" or "Sleepy!" or "Very sleepy!"



HelloTrainees.cpp

```cpp
#include <iostream>

using namespace std;

int main(){
    int hours_slept;
    cin >> hours_slept;
    if (hours_slept > 8){
        cout << "Awake!";
    } else if (hours_slept == 8){
        cout << "Sleepy!";
    } else {
        cout << "Very sleepy!";
    }
    return 0;
}
//Note the brackets/braces and the ==.
```

5. Relational & logical operators

If the relation is **true**, it returns **1** whereas if the relation is **false**, it returns **0**.

Operator	Meaning	Example
`==`	Is Equal To	`3 == 5` gives us **false**
`!=`	Not Equal To	`3 != 5` gives us **true**
`>`	Greater Than	`3 > 5` gives us **false**
`<`	Less Than	`3 < 5` gives us **true**
`>=`	Greater Than or Equal To	`3 >= 5` give us **false**
`<=`	Less Than or Equal To	`3 <= 5` gives us **true**

Good resource.

5. Relational & logical operators

Operator	Example	Meaning
&&	expression1 **&&** expression2	Logical AND. True only if all the operands are true.
||	expression1 **||** expression2	Logical OR. True if at least one of the operands is true.
!	**!**expression	Logical NOT. True only if the operand is false.

Good resource.

Hands-on practice time! (Again)

Problems discussed by mentors were:

1- Given 4 numbers, print the maximum out of them.

Input: four integer numbers.
Output: one number, their maximum.

Solutions in next slide. Please try your best before moving on to the solution.

Problems discussed by mentors were:

```cpp
#include <iostream>
#include <limits.h>
//We include <limits.h> to be able to use LLONG_MIN
using namespace std;

int main(){
    long long n1, n2, n3, n4;
    long long mx = LLONG_MIN;
    cin >> n1;
    if (n1 > mx) mx = n1;
    cin >> n2;
    if (n2 > mx) mx = n2;
    cin >> n3;
    if (n3 > mx) mx = n3;
    cin >> n4;
    if (n4 > mx) mx = n4;

    cout << mx;

    return 0;
}
```

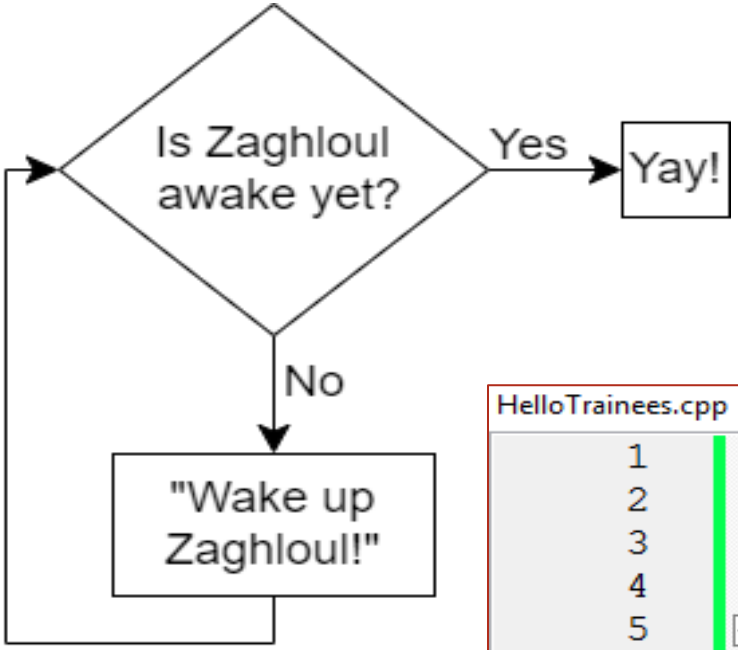
OR

```cpp
#include <iostream>
using namespace std;

int main(){
    long long mx, n2, n3, n4;
    cin >> mx >> n2;
    if (n2 > mx) mx = n2;
    cin >> n3;
    if (n3 > mx) mx = n3;
    cin >> n4;
    if (n4 > mx) mx = n4;

    cout << mx;

    return 0;
}
```

05. B)
Control flow:
A Newer problem

Oh no! Zaghloul forgot to set his alarm!

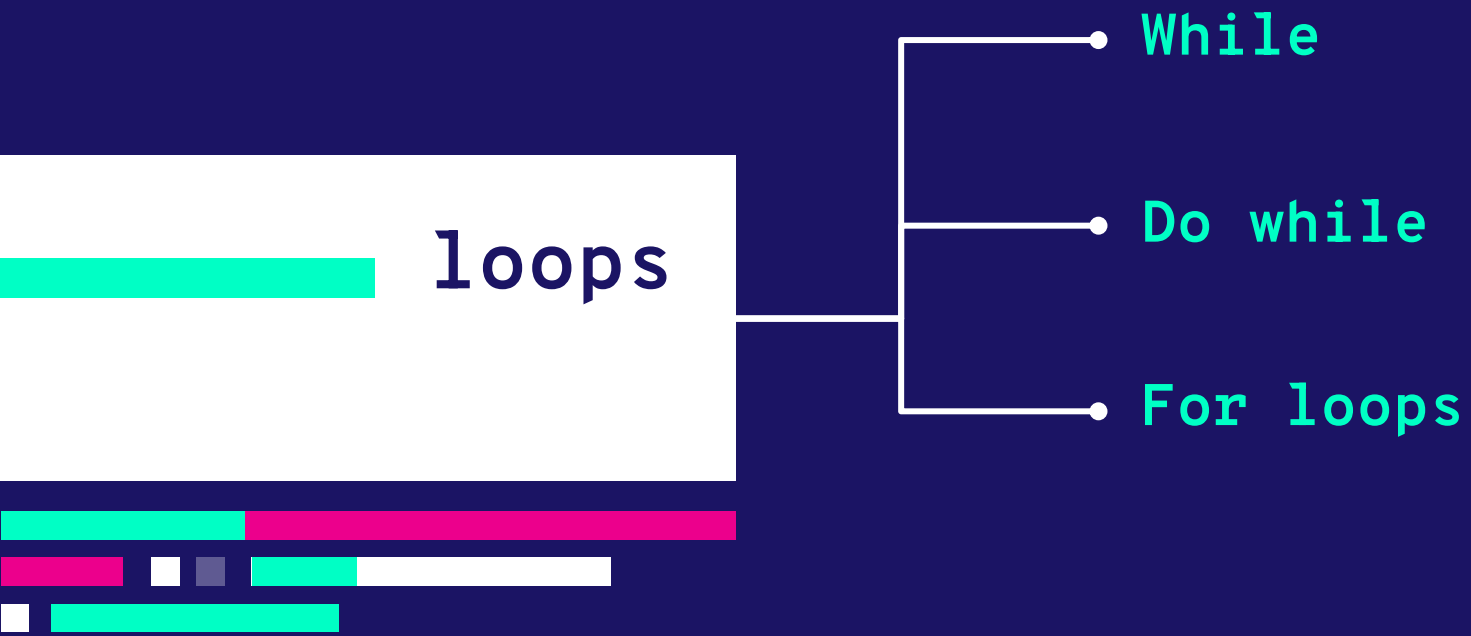
Keep yelling at Zaghloul until he wakes up!



HelloTrainees.cpp

```cpp
#include <iostream>

using namespace std;

int main(){
    bool zaghloul_is_asleep = 1;
    while (zaghloul_is_asleep){
        cout << "Wake up zaghloul!" << endl;
        cin >> zaghloul_is_asleep;
    }
    return 0;
}
//Note the endl and the omission of ==.
```

How can we repeat code?



While loop

```
int x=1;
while(x<=10){
    cout<<x++<<" ";
}
cout<<endl;
```

Do while loop

```
int x=1;
do{
    cout<<x++<<" ";
} while (x<=10);
cout<<endl;
```

For loops

```
int main(){
    for (int i = 0; i < 10; i ++){
        cout << i << endl;
    }
    return 0;
}
```

Mostly used when repeating a block of code a known amount of time.

Hands-on practice time! (III)

Problems discussed by mentors were:

Given the function $Y = X^3$, print the sequence of the output from $X = L$ till $X = R$.

Input: Two integers **L** and **R**, where $0 \leq L \leq R \leq 4000$.
Output: The sequence of the outputs Y, where each number is on a separate line.

Solution in next slide. Please try your best before moving on to the solution.

Problems discussed by mentors were:

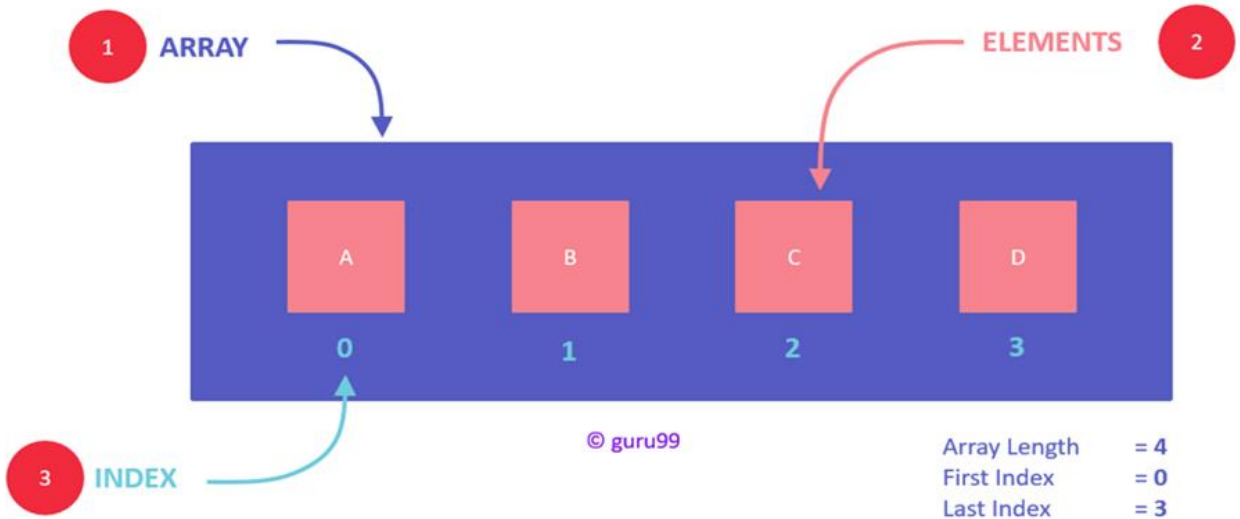
```cpp
#include <iostream>
using namespace std;

//You MUST be careful of the input limits to this problem
//Input could reach 4000, so output could reach 4000^3
//That's 6.4*10^10 !! A HUGE number.
//so using int won't suffice.

//The i must be long long as well, bc. of the same problem
//we faced in slide 23. If the operands around * are of the type
//int, then the result will be calculated as int before being assigned
//to res. so we either do this (long long) i * i * i or just have i as
//long long from the start
int main(){
    int L, R;
    cin >> L >> R;
    for (long long i = L; i <= R; i++){
        long long res = i * i * i;
        cout << res << endl;
    }
    return 0;
}
```

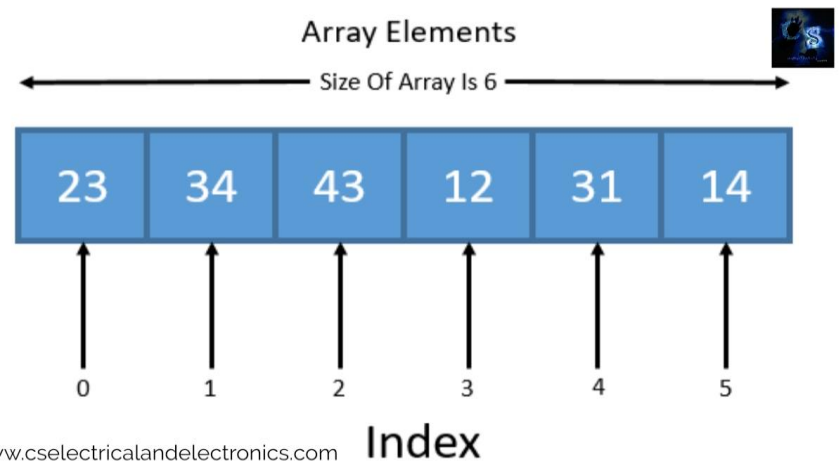
06. Arrays & Strings

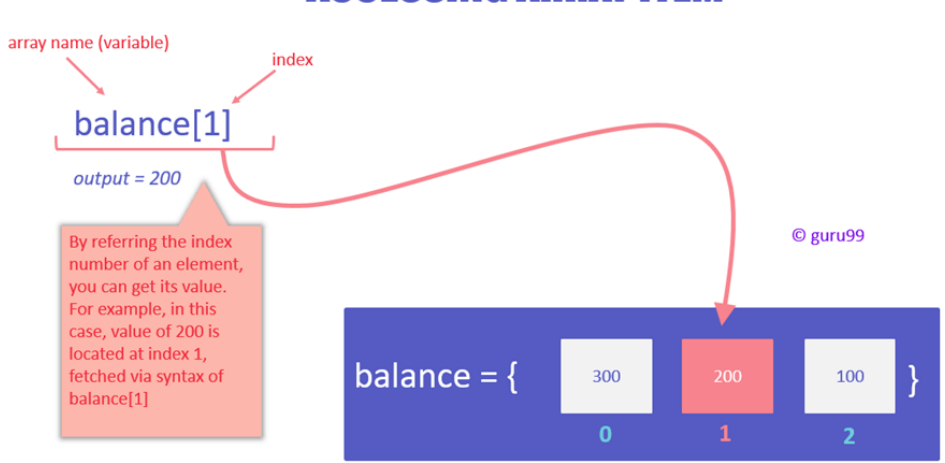
Zaghloul asked ten of his friends for their feedback on their last ASUFE CPC session, and each gave him a rating from 1 to 5.

He wants you to print the numbers he recorded.

Concept Diagram



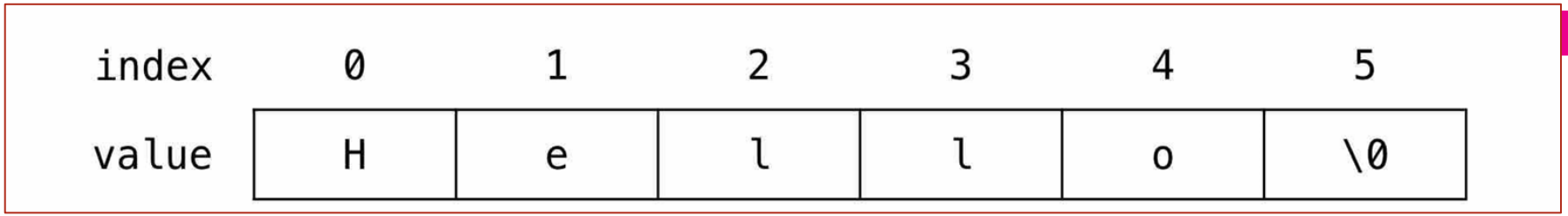




```cpp
#include <iostream>

using namespace std;

int main(){
    int feedbacks[10];
    for (int i = 0; i < 10; i++){
        cin >> feedbacks[i];
    }
    for (int i = 0; i < 10; i++){
        cout << feedbacks[i] << ' ';
    }
}
//Note the 0-indexing and the square brackets.
```

```
4 4 5 3 5 1 5 5 4 3
4 4 5 3 5 1 5 5 4 3
Process returned 0 (0x0)
Press any key to continue.
```

```
string s;
s = "Hello";
```

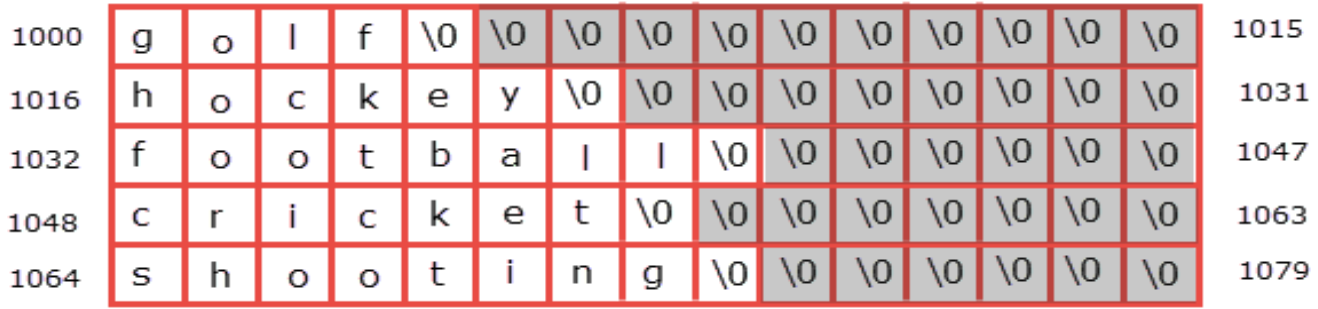


But, what if we want to store multiple names in an array?

An array of arrays!

```
string sports[5];
sports[0] = "golf";
```

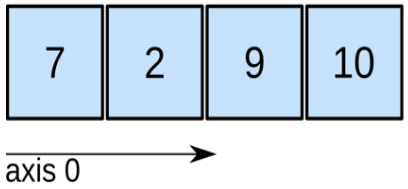
sports[5][15]



Memory representation of an array of strings or 2-D array of characters

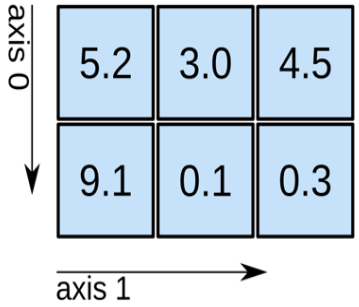
TheCguru.com

1D array



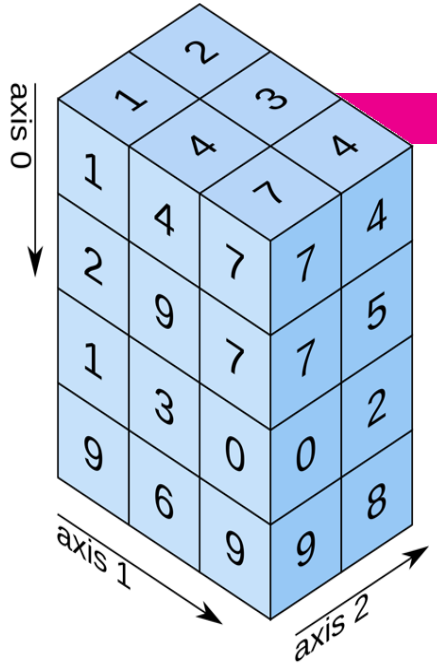
shape: (4,)

2D array



shape: (2, 3)

3D array



shape: (4, 3, 2)

Hands-on practice time! (IV)

Problems discussed by mentors were:

1- Given an array of integers of size n, print the summation of numbers between the indices **R** and **L**.

Input: Three integers **n**, the size of the array, **R** and **L,** where $0 \leq L \leq R < n$, and $1 \leq n \leq 1000$, followed in a new line by **n** integers where each number in the array is smaller than 10^9 and bigger than 0.

Output: A single integer, sum of numbers between the indices **L** and **R** inclusive.

Solution in next slide. Please try your best before moving on to the solution.

Problems discussed by mentors were:

```cpp
#include <iostream>
using namespace std;

//You MUST be careful of the input limits to this problem
//A single number in the array could reach 10^9, and
//array size could reach 1000, so if all numbers in the
//array were to be 10^9, and if L was 0 and R was n,
//the resulting sum would be 10^9 * 10^5 which CANNOT
//fit in an int.
int main(){
    int n, L, R;
    long long arr[1005];
    cin >> n >> L >> R;
    //I started from 1 to make things easier in case
    //L was 0.
    arr[0] = 0;
    for (int i = 1; i <= n; i++){
        cin >> arr[i];
        arr[i] += arr[i-1];
    }
    cout << arr[R+1] - arr[L];
    //R+1 here because my array actually starts at index 1, remember?
    return 0;
}
```

RECAP

PROBLEM SOLVING & IDE

a. What is problem solving?
b. Where to write your code

DATA TYPES AND OPERATORS

a. What are the different data types out there?
b. What can you do with them? (how to operate on them?)

IF ELSE STATEMENTS

a. What if there are different scenarios?
b. Relational operators

LOOPS

a. What if you need to repeat your code?

ARRAYS

a. What if you have too much data to name every single variable?
b. What if your data has multiple dimensions?

Resources

Links to videos that explain this session's content: ASUFE-CPC [1] OR ACM ASCIS [1, 2, 3].

Links to a few problem-solving sites: Codeforces, LeetCode, HackerRank, Timus, AtCoder.

Links to C++ docs (will be very important later): cppreference, cplusplus.

Links to useful sites in general: cp-algorithms, GeeksForGeeks, Programiz, TutorialsPoint.

An amazing channel you MUST check out: Arabic Competitive Programming

Complete roadmaps to problem solving: one, two, three.

Good luck!

THANKS !

Do you have any questions?
Send on Discord