



acmASCIS
spreading knowledge

AGENDA

- The super equation ☺
- Challenge & quick review on functions
- String
 - what ?
 - how ?
 - why ?
- ASCII code?
- String functions :
- Competition
- The big challenge [big integer]

The super equation

علم + عمل + أخلاق = نهضة



م =	Ж
ى =	Ч
ن =	Ю
ش =	Ң
ر =	Ң
ب =	Ә
ا =	И
ل =	С
؟ =	Г
س =	Д
ك =	□
ف =	▲
ت =	●
ع =	П
ه =	Ф
ط =	Ү
و =	Ҽ
ح =	Ӯ
ـ =	Ө



ХД △ Π
ΔЖЮΣΠ



ASCII table

Dec	Hx	Oct	Char		Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0 000	000	NUL	(null)	32	20	040	 	Space	64	40	100	@	Ø	96	60	140	`	^
1	1 001	001	SOH	(start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2 002	002	STX	(start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3 003	003	ETX	(end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4 004	004	EOT	(end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5 005	005	ENQ	(enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6 006	006	ACK	(acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7 007	007	BEL	(bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8 010	010	BS	(backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9 011	011	TAB	(horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A 012	012	LF	(NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B 013	013	VT	(vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C 014	014	FF	(NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D 015	015	CR	(carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E 016	016	SO	(shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F 017	017	SI	(shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10 020	020	DLE	(data link escape)	48	30	060	0	Ø	80	50	120	P	P	112	70	160	p	p
17	11 021	021	DC1	(device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12 022	022	DC2	(device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13 023	023	DC3	(device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14 024	024	DC4	(device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15 025	025	NAK	(negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16 026	026	SYN	(synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17 027	027	ETB	(end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18 030	030	CAN	(cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19 031	031	EM	(end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A 032	032	SUB	(substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B 033	033	ESC	(escape)	59	3B	073	;	:	91	5B	133	[[123	7B	173	{	{
28	1C 034	034	FS	(file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D 035	035	GS	(group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E 036	036	RS	(record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F 037	037	US	(unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Null character

' \0'

,/0,



```
int main()
{
    char x = 20, y = '8';

    cout << x << endl;
    cout << y + 1 << endl;

    return 0;
}
```

H

9

Challenge



Functions for character array

- 1) Get the length / size :-**
- 2) Add new character to the array end :-**
- 3) Insert new character into the array**
- 4) Erase a character from the array :-**
- 5) Find if a character is included or not**
- 6) Clear the array :-**

Get the length / size :-

```
int size(char arr[])
{
    int counter = 0;
    while ( arr[counter] != '\0' )
        ++ counter;

    return counter;
}
```

```
int main()
{
    cout << size("ACM_ascis") << endl
    return 0;

}
```

Erase a character from the array :-

```
void Erase (char arr[], int index)
{
    for(int i = index; arr[i] != '\0'; ++i)
    {
        arr[i] = arr[i + 1];
    }
}
```

Find if a character is included or not

```
int Find(char arr[] , char target)
{
    for(int i = 0; arr[i] != '\0' ; ++i)
    {
        if(arr[i] == target)
            return i ;
    }
    return -1;
}
```

Insert new character to the array :-



This is hard !!!!!



STRING

What



Why

HOW

#include <string>

□ **Declaration :**

string str;

□ **Initialization :**

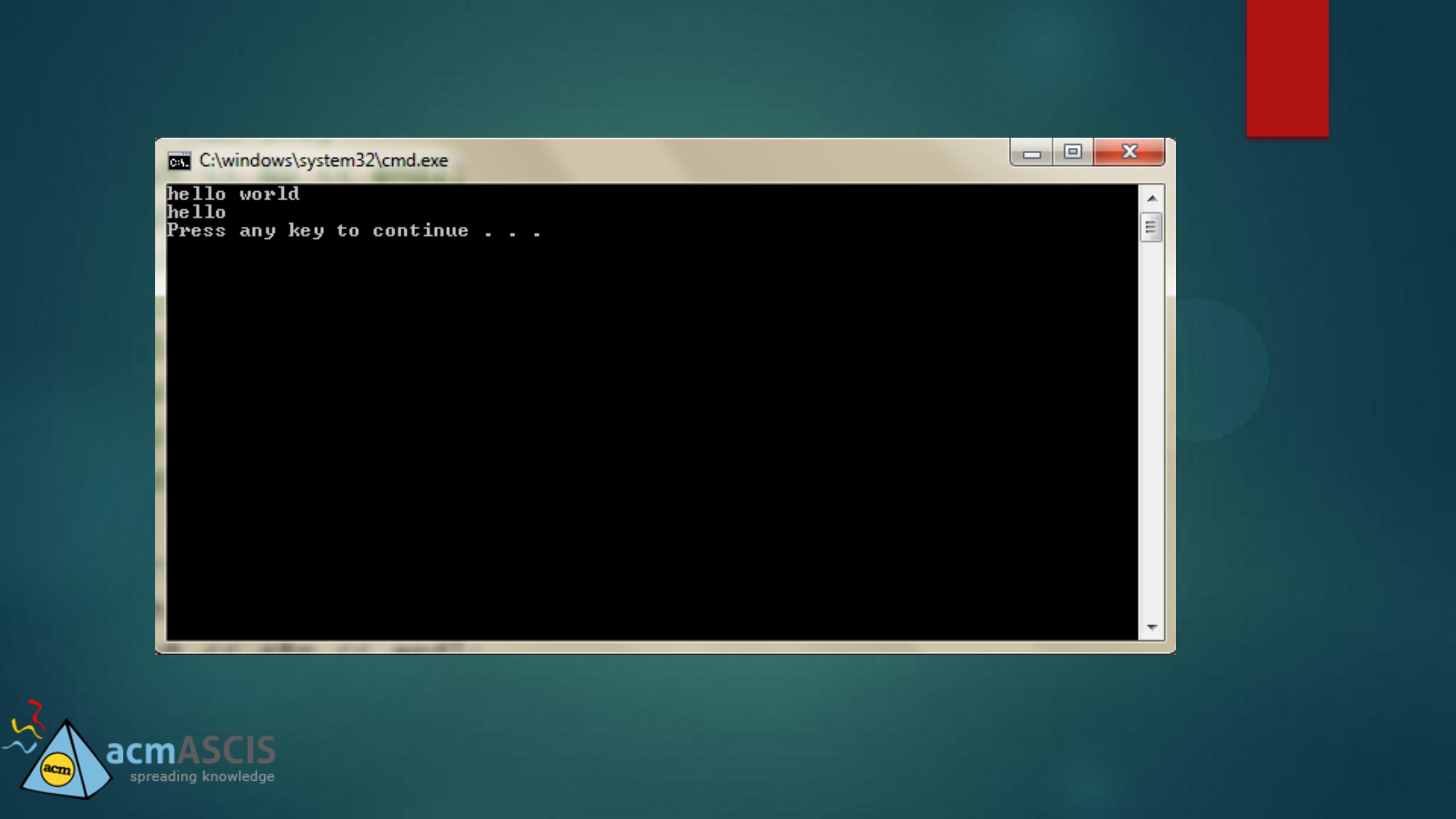
string str = “Hello”;

□ **I/O :**

cin >> str ;

cout << str << endl;

```
C:\windows\system32\cmd.exe
hello
hello
Press any key to continue . . .
```



A screenshot of a Windows command prompt window titled "cmd C:\windows\system32\cmd.exe". The window contains the following text:
hello world
hello
Press any key to continue . . .



getline (cin, string);

```
string str;
```

```
getline(cin, str);
```

C:\> C:\windows\system32\cmd.exe

Hello World
Hello World
Press any key to continue . . .

Random access operator

[]

```
string str = "start";
cout << str[2] << endl;
```

[2]
↓

String concatenation !!!

```
String s, d;
```

```
s = "fcis", d = "asu";
```

```
s += d;
```

```
Cout << s << endl;
```

What about mathematical operations ?

```
string str = "500";  
cout << str /2 << endl;
```



Access operator

```
String str ;
```

Str .



append

size

clear

insert

find

erase

substr

Size

```
String s1 = "Hello", S2 = "Hello World" ;
```

```
Cout << s1.size()<<endl;
```

5

```
Cout << S2.size() <<endl;
```

11

Find

Str.find (char) | | Str.find (string)

```
String s1 = "Hello", s2 = "Hello World" ;
```

```
Cout << s2.find('W') << endl;
```

```
Cout << s1.find('W') << endl;
```

6

-1

Erase

Str.erase(position, counter)

String S2 = “Hello World” ;

Cout << s2.erase(5, 6)<<endl;

Erase

Str.erase(position, counter)

String S2 = “Hello World” ;



Cout << s2.erase(5, 6)<<endl;

Erase

Str.erase(position, counter)

String S2 = “Hello World” ;

Cout << s2.erase(5, 6)<<endl;

Hello

insert

`str.insert(pos, string) || str.insert(pos, freq, char)`

`String str1 = "My is basem", str2 = "inert";`

```
str1.insert(3, "name ");
str2.insert(2, 1, 's');
```

insert

`str.insert(pos, string) || str.insert(pos, freq, char)`

`String str1 = "My is basem", str2 = "inert";`

`str1.insert(3, "name ");`

`str2.insert(2, 1, 's');`

insert

`str.insert(pos, string) || str.insert(pos, freq, char)`

`String str1 = "My is basem", str2 = "inert";`

```
str1.insert(3, "name ");
str2.insert(2, 1, 's');
```

insert

My name is
basem

Substr

str.substr(start,size);

```
String str = "Hello world" ;  
str = str.substr(6,5) ;
```

world

pop_back

`str.pop_back();`

```
String str = “world” ;  
str = str.pop_back() ;
```

world

Note : this is not a standard function for string class
uva wont accept your code if it containing this function

push_back

str.push_back(char);

```
String str = "worl" ;  
str = str.push_back('d') ;
```

world


```
string x ="str" ;  
x.pop_back();  
x.push_back('c');  
x.insert(0,1,'a') ;  
x.erase(2,1);  
x.append(1,'m');  
x.erase(1,1);  
cout << x << endl ;
```

The big challenge

If

a = “12313454534345333543543453435435354”

B = “5646543554354345343543543543545435435”

Print

A + b

A * b

a / b

Sqrt(a)



Any question ?

Thank you ☺