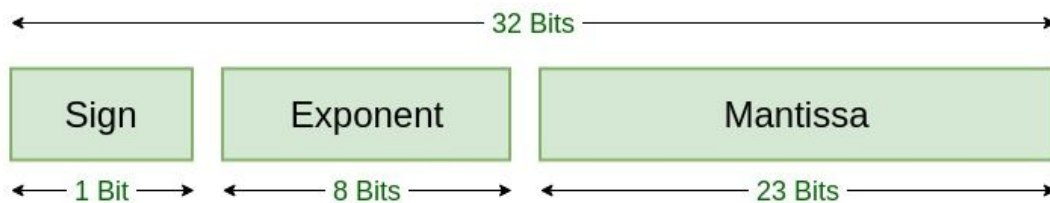


## Упражнение 2

Типове данни и тяхното представяне. ASCII таблица.

- bool, char, int, float, double.
- short, long, unsigned, signed - **модификатори**.
- Стандарта за **размера** на типовете:  
 $|short| \leq |long| \leq |long long|$ , **НО**  $|short| < |long long|$   
 $|char| \leq |short|$ , **НО**  $|char| < |int|$
- **Представяне** на типа **float** в паметта:



### Single Precision IEEE 754 Floating-Point Standard

- **ASCII таблица** - всеки **char** има стойност от 0 до 255, която представлява символ (по подразбиране **char** е *unsigned*).

## ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

## Думи в C++.

- **Запазени думи** - всички имена на типове, *if*, *else*, *for* и др.  
<https://en.cppreference.com/w/cpp/keyword>
- **Идентификатори** - малки и главни букви, цифри и долни черти, но **не** започват с цифра.
- **Литерали**

Примери:

- 2, -5 - литерали от тип *int*.
- 2.5, 2.0, 2. - литерали от тип *double*.
- 6.3f, .6f, - литерали от тип *float*.
- 'A', '5', 'c' - литерали от тип *char*.
- false - литерал от тип *bool*.
- 5u - литерал от тип *unsigned int*.
- 7ul - литерал от тип *unsigned long int*.
- 0123 - литерал от тип *int* в **осмична** бройна система.
- 0x1B9 - литерал от тип *int* в **шестнадесетична** бройна система.

## Променливи и област на действие.

- **Декларация и дефиниция** - създаване, даване на стойност.
- **Област на действие** - част от кода, където променливата "живее" (е видима).
- **Преобразуване (кастване)** - смяна на начина по който интерпретираме данните.

- ([тип])[променлива];

Пример: *int* a = 1500 000 000, b = 1521234 654;

*long long* c = a + (*long long*)b;

/\* Интерпретираме променливата **b** като *long long*, събираме с **a** и присвояваме сбора в **c**.

Забележка: Резултатът от израза

**a** + (*long long*)**b**, е от тип *long long*, защото събираме променлива от тип *int* с

променлива от тип *long long* (резултатът е от по-големия тип). \*/

## Оператори.

- **Аритметични** - (+, -, \*, /, %, ++, --) събиране, изваждане, умножение, деление, **остатък от деление**, **инкрементиране** (увеличаване с 1), **декрементиране** (намаляване с 1).
  - Целочислено деление.
    - `float c = 5 / 3;`      `// c = 1`  
`// Забележка: Резултатът от деление на две`  
`// целочислени числа е целочислено число.`
  - Деление по модул (остатък от деление).
    - `float d = 5 % 3;`      `// d = 2`
  - Дробно деление.
    - `float e = 5 / 3.0f;`      `// e = 1.6666...`
- **Оператори за сравнение** - (>, <, >=, <=, ==, !=) по-голямо, по-малко, по-голямо или равно, по-малко или равно, **равно**, **различно**.
- **Логически** - (&&, ||, !) логическо “и”, “или”, **отрицание (негация)**.
- **Побитови** - (&, |, ^, ~) побитови “и”, “или”, “**изключващо или**”, **отрицание (негация)**. Действат върху отделните битове на променливите (двоичния запис на променливата).  
**Побитовите операции не работят над типове с плаваща запетая!**
- **Съкратен запис** (+=, -=, \*=, /=, %=, &=, |=, ^=)
  - `a (знак) = b;` ⇔ `a = a (знак) b;`  
*Пример: `a += b` ⇔ `a = a + b;`*
- **Оптимизацията на логическите оператори**
  - Изрази от **типа** на `false` && (“**други неща**”).
  - Изрази от **типа** на `true` || (“**други неща**”).
  - “Другите неща” може изобщо да не бъдат сметнати и **първият** израз да се изчисли на `false` без да се проверят “**другите неща**”, а **вторият** - `true` без да се проверят “**другите неща**”.

a	b	a && b	a    b	!a
false	false	false	false	true
false	true	false	true	
true	false	false	true	false
true	true	true	true	

a	b	a & b	a   b	~a	a ^ b
1010	1110	1010	1110	0101	0100
1001	0101	0001	1101	0110	1110

a	b	a == b	a != b	a < b	a <= b	a > b	a >= b
10	14	false	true	true	true	false	false

### Части на програмата.

- **Заглавни файлове** - `#include <файл>` и `#include "файл"`.
- **Блокове** - част от кода ограден от { и }.
- **main()** - стартовата точка на програмата.
- **Коментари** - текст, който **няма да бъде** компилиран/изпълнен, пише се от програмисти за програмисти. Най-често е допълнителна информация за това как работи програмата или за какво се използва дадена променлива и др.
  - `//` Едноредов коментар.
  - `/*` Многоредов коментар. `*/`

### Потоци за вход и изход.

- Чрез **cout** и оператора `<<` пишем в конзолата.
- Чрез **cin** и оператора `>>` четем от конзолата.

### Приоритет на операциите и асоциативност.

[https://en.cppreference.com/w/cpp/language/operator\\_precedence](https://en.cppreference.com/w/cpp/language/operator_precedence)

Скобите са хубаво нещо, ползвайте ги. :)

## Integrated Development Environment (IDE).

- Работна среда предлагаща *текстов редактор*, компилатор, свързващ редактор (linker) и допълнителни *глезотийки*, с които да улесни писането на код.

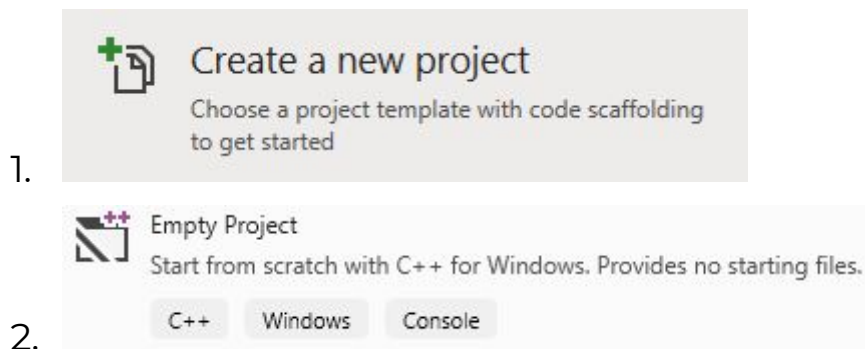
- **Инсталиране на Visual Studio 2019 (от ELMS).**

<http://elms.fmi.uni-sofia.bg/>

По време на инсталацията от списъка с компоненти изберете *Desktop development with C++*.



- **Създаване на проект във Visual Studio 2019**



- **Създаване на файл в проект**

В прозореца *Solution Explorer* чрез натискане на **десен бутон** върху *Source Files* → *Add* → *New Item* → *C++ File (.cpp)*

- **Стартиране на програмата**

*Debug* → *Start Without Debugging* (**Ctrl + F5**)

- **Добавяне на Breakpoint**

Използва се за “*паузиране*” на изпълнението на програмата по време на *Debugging*.

Добавя се чрез натискане на тъмната лента от ляво на номера на реда.

- **Стартиране на Debugger**

*Debug* → *Start Debugging* (**F5**)

- **Изпълняване на текущ ред по време на Debugging**

Чрез бутона *Step Over* (**F10**) 

- **Спиране на Debugger**

Чрез бутона *Stop Debugging* (**Shift + F5**) 