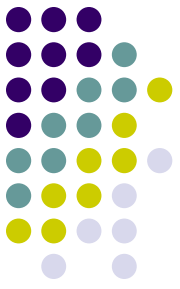


Введення-виведення інформації

Тема 4





Стандартне введення

- Файл заголовків `<stdio.h>`
- ***scanf("формати", адреси змінних)***

Приклад.

```
{int x;  
float a;  
scanf("%d%f", &x, &a);  
}
```

Scanf("\n"); - *очистка буфера перед введенням символної інформації*

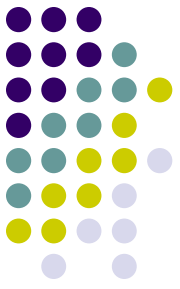
Стандартне введення



- **Формати**

%i %d	10-і цілі числа
%u	Беззнакові 10-і числа
%o	Числа в 8 сист. счисл.
%x	числа в 16 сист. счисл.
%f	float
%e %E	з плаваючою точкою (звичайна точність [-]d.ddde+dd;)

%g %G	З плаваючою точкою (подвійна точність)
%c	СИМВОЛИ
%s	рядки
%p	показчики
%lf	double
%ld	long



Стандартне введення

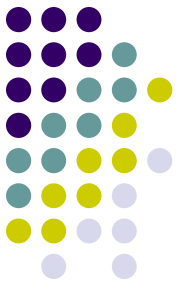
- *getchar()*

Приклад.

```
#include <stdio.h>  
#include <conio.h>
```

- *getch()*
{conio.h}

```
main( )  
{ char ch;  
  ch=getchar( );  
  getch();  
}
```



Стандартне виведення

- Файл заголовків `<stdio.h>`
- ***printf*** (“шаблон”, список змінних)

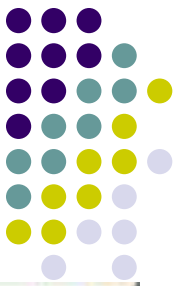
Приклад.

```
printf (“При  $x=%d$   $y=%f$ ”,  $x$ ,  $y$ );
```

Обмеження знаків, що виводяться

%5.3f

Стандартне виведення

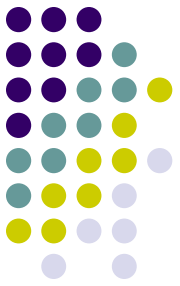


```
c:\VCPP\BIN\VC.EXE
-0.256000
-0.256
-0.256
      -0.256
```

```
c:\VCPP\BIN\VC.EXE
E File Edit Search R
[ ]
#include <iostream.h>
#include <stdio.h>
#include <conio.h>

void main()
{
clrscr();
float f=-0.256f;

printf("%f",f);
printf("\n%6.3f",f);
printf("\n%5.3f",f);
printf("\n%12.3f",f);
}
```



Стандартне виведення

- ***putchar()*** друкує один символ

Пример.

```
putchar('D');
```

```
putchar("\n");
```

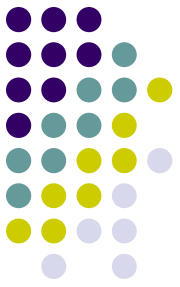
```
putchar("\007");
```

```
putchar(ch); /* змінна типу char */
```

```
putchar(getchar( ));
```

- ***puts()*** - друкує рядок

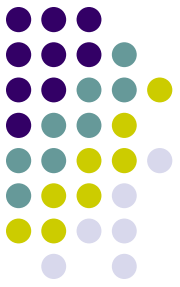
Потокове введення- виведення



- Файл заголовків <iostream.h>
- ***istream*** – потік введення
Об'єкт cin
>>-операція введення (повертає посилання на потік введення)
- ***ostream*** – потік виведення
Об'єкти cout та cerr
<<-операція виведення

Для всіх вбудованих типів

Потокове введення- виведення



Приклади.

```
int x; cin >> x;
```

```
cout << "\tПриклад виведення: " << 34;
```

```
cout << " Результат: x=: " << x <<" \n" ;
```

Потокове введення- виведення



- Маніпулятори і форматоване введення-виведення

endl	при виведенні перейти на наступний рядок;
ends	вивести нульовий байт (признак кінця рядка символів);
flush	без затримки вивести всю інформацію і очистити буфери обміну;
dec	вводити-виводити числа в десятковій системі (за замовчуванням);
oct	вводити-виводити числа в восьмеричній системі;
hex	вводити-виводити числа в шістнадцятиричній системі;

Потокове введення- виведення



- Маніпулятори і форматоване введення-виведення
 - `io manip.h`

`setw (int n)`

Виводити число в задану кількість знакомісць (встановити ширину поля);

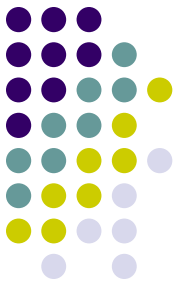
`setprecision (int n)`

встановити кількість знаків після коми для дійсних чисел;

`setbase (int n)`

встановити систему счислення для виведення чисел; `n` може приймати значення 0, 2, 8, 10, 16, (0 означає систему счислення за замовчуванням, тобто 10);

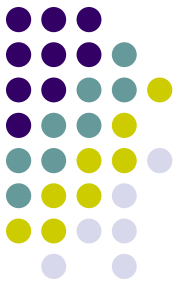
Потокове введення- виведення



- Маніпулятори і форматоване введення-виведення

<code>ws</code>	перемикає потік введення в такий режим, при якому всі пробіли, табуляція, переведення рядка і т.і. будуть вводитися. За замовчуванням вони сприймаються як роздільники між атрибутами введення
<code>setfill(int n)</code>	встановити символ-заповнювач; цим символом значення, що виводиться, буде доповнюватися до потрібної ширини;

Потокове введення- виведення



Приклади.

```
int x = 53;
```

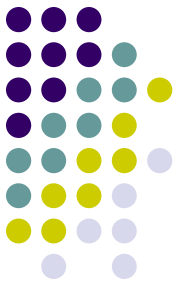
```
cout << "Десятковий вид: " << dec << x << endl  
    << "Восьмеричний вид: " << oct << x << endl  
    << "Шістнадцятеричний вид: " << hex << x << endl;
```

```
float y=1.256f;
```

```
cout << setw(6) << setprecision(2) << y << endl;
```

```
int x;
```

```
cin >> hex >> x;
```



Поток введення

- `cin >>x1 >>x2 >>x3;`
`((cin >>x1) >>x2) >>x3;`
- `cin.get()` – введення одного символу, який повертається як результат
`cin.get() >>x1 >>x2 ; ?`
- `cin.get(x);` - повертає покажчик на потік введення
`cin.get(a).get(b).get(c); cin.get(a) >> b;`