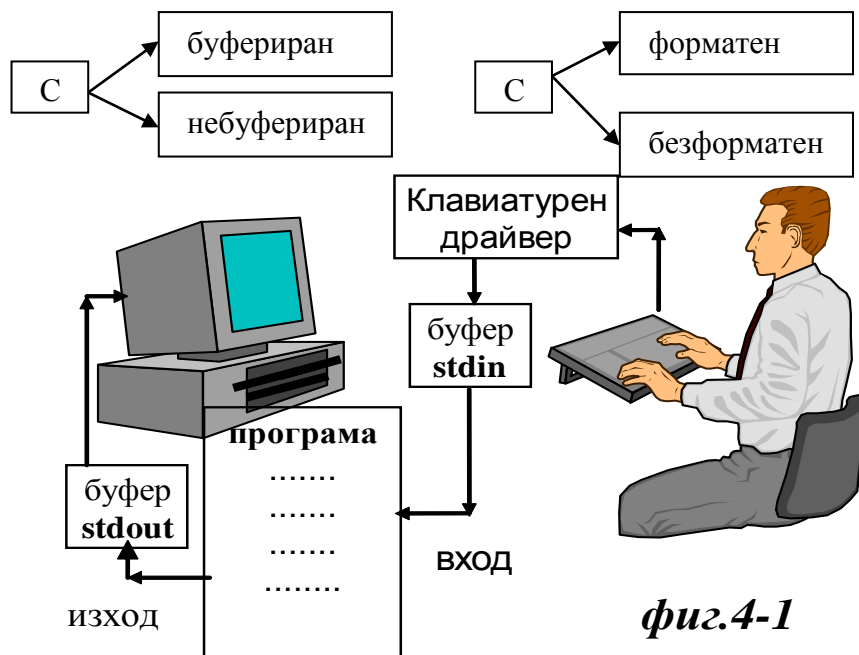


лабораторно упражнение № 4

Понятие за вход/изход в езика C (I/O)

времетраене: 2 уч. часа

4. Вход/изход (I/O) – файлов обмен



- **stdin** - входен файл, свързан със стандартния вход
- **stdout** - изходен файл, свързан със стандартния изход
- **stderr** - изходен файл за извеждане на системните грешки

Вход/изходът се реализира с функции, описани в стандартните *header* файлове: **stdio.h** и **conio.h**

4.1 Буфериран форматен вход/изход

За реализиране на този тип обмен в *header* файла **stdio.h** съществуват двойката функции *scanf()* и *printf()*. И двете функции са с променлив брой на аргументите, като винаги първият аргумент е константен символен низ, който се нарича *форматиращ* низ. Чрез него се специфицира начина на обмен с входно/изходните канали.

структура на форматиращите последователности:

% флагове ширина . точност F

→ → →

прототип на функция за буфериран форматен изход:

```
int printf( const char *format [, argument]... );
```

прототип на функция за буфериран форматен вход:

```
int scanf( const char *format [,address_of_prgument]... );
```

забележки:

- броят на форматните спецификации задължително трябва да е равен на броя на аргументите;
- функциите връщат броя на обменените *argument*;

4.2 Буфериран безформатен вход/изход

прототип на функция за буфериран безформатен изход:

```
int putchar( int c );
```

функцията връща като резултат ASCII кода на символа, които се извежда, при грешка – EOF

прототип на функция за буфериран безформатен вход:

```
int getchar( void );
```

функцията връща като резултат ASCII кода на символа, които се прочита от клавиатурния буфер или EOF при край на файла (cntrl/Z)

4.3 Небуфериран безформатен вход

За реализиране на този тип обмен в *header* файла **conio.h** съществува двойката функции `_getc()` и `_getche()`

```
int _getc( void );
```

```
int _getche( void );
```

4.4 Пример за тестване на форматни спецификации при извеждане на щели (*int*) стойности

```
#include <Visual_2010.h>
#include <stdio.h>
#include <conio.h>

int main (void)
{
    int i, j;
    /*-----*/
    i=567; j=-234;

    printf("\n format specifications for int values in language C");
    printf("\n-----");
    printf("\n\n printing variables  i=567,j=234\n\n");
    printf("\nformat specification  %cd  and  %c6d  ->|%d|%d|%6d|%6d|, '%','%'.i,j, i, j);
```

```

printf("\nformat specification %c.6d %c1d ->|%%.6d|%.6d|%.1d|%.1d|", '%', '%', i, j, i, j);
printf("\nformat specification %c+6d ->|%.6d|%.6d|", '%', i, j);
printf("\nformat specification %c-6d ->|%.6d|%.6d|", '%', i, j);
printf("\nformat specification %c+06d ->|%.6d|%.6d|", '%', i, j);
printf("\nformat specification %c-06d ->|%.6d|%.6d|", '%', i, j);
printf("\nformat specification %c+.06d ->|%.6d|%.6d|", '%', i, j);
printf("\nformat specification %c4x ->|%.4x|%.4x|", '%', i, j);
printf("\nformat specification %c4X ->|%.4x|%.4x|", '%', i, j);
printf("\nformat specification %c#4x ->|%.4x|%.4x|", '%', i, j);
printf("\nformat specification %c#04x ->|%.04x|%.04x|", '%', i, j);
printf("\nformat specification %c#6o ->|%.6o|%.6o|", '%', i, j);
printf("\nformat specification %c#6u ->|%.6u|%.6u|", '%', i, j);

```

```

printf("\n\nPress any key to continue");
_getch();
return 0;
}

```

```

изход на програмата
format specifications for int values in language C
-----
printing variables i=567,j=234

format specification %d ->|567|-234|_ _ _ 567|_ _ -234|
format specification %.6d ->|000567|-00234|567|-234|
format specification %+6d ->|_ _ +567|_ _ -234|
format specification %-6d ->|+00567|-234_ _ |
format specification %+06d ->|+00567|-00234|
format specification %-06d ->|567_ _ _ |-234_ _ |
format specification %+.06d ->|+000567|-000234|
format specification %4x ->|_ 237|ff16|
format specification %4X ->|_ 237|FF16|
format specification %#4x ->|0x237|0xff16|
format specification %#04x ->|0x237|0xff16|
format specification %#6o ->|_ 01067|0177426|
format specification %6u ->|_ _237|_ 65302|

Press any key to continue
забележка: В изхода празните интервали за посочени като _

```

задания за самостоятелна работа(разклонени алгоритми):

1. Да се доразвие програмата, като се допишат варианти на форматни спецификации за **float** променливи.

2. Да се състави блоков алгоритъм и C-програма за :

Да се състави блоков алгоритъм и C- програма за пресмятане на функцията $y=f(x)$:

$$y = \begin{cases} b+x & \text{ако } x \leq m1 \\ a*x & \text{ако } m1 < x \leq m2 \\ 4-x & \\ c+x & \text{ако } x > m2 \end{cases}$$

Входни данни са: a,b,c m1,m2(цели стойности), които се въвеждат от клавиатурата



ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ

ФАКУЛТЕТ ПО КОМУНИКАЦИОННА ТЕХНИКА И ТЕХНОЛОГИИ

Лабораторно упражнение No 4

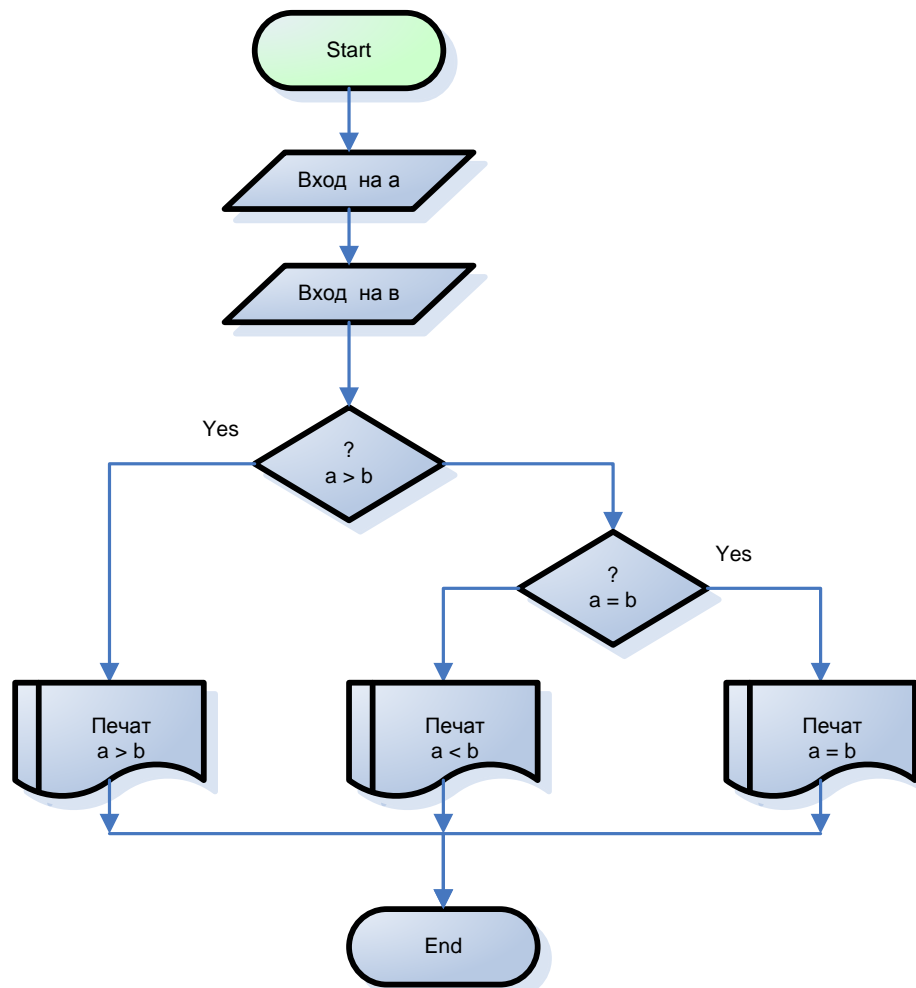
задание: Да се състави C-програма за обработка на две цели стойности, постъпващи от клавиатурата, като на екрана се изведе оценката им по големина.

студент:
фак. номер:
група

преподавател:
подпис:

дата: xx.xx.xxxx г.

гр.София
блоков алгоритъм:



сурс код: C –програма

```
#include "stdafx.h"
```

```
int main(int argc, char* argv[])
{
    int a, b;
    //-----
    printf("\n input integer value a=");
    scanf("%d", &a);
    printf("\n input integer value b=");
    scanf("%d", &b);
    if(a>b)
        printf("\n a=%d, b=%d -> a>b \n",a,b);
    else
        if(a==b)
            printf("\n a=%d, b=%d -> a=b \n",a,b);
        else
            printf("\n a=%d, b=%d -> a<b \n",a,b);
    return 0;
}
```