

Технически Университет – филиал Пловдив
Факултет по Електроника и Автоматика

Протокол №:1

Тема: Работа с printf() и scanf().

Изготвил: Даниел Райчев Славчев

Факултетен номер: 382447

Специалност: КСТ

Дата: 12.10.2023г.

Група: 42б

1) Въпроси

1. Кои свойства на програмния език C го правят подходящ за решаване на разнообразни инженерни задачи?

- C е подходящ за разнообразни инженерни задачи благодарение на близостта до хардуера, портатилността, ефективността, богатите библиотеки, възможността за мултиплатформено програмиране и голямата общност от програмисти.

2. В какъв тип файлове се съхраняват програмите на C?

- Програмите на C се съхраняват в текстови файлове, които обикновено имат разширение ".c". Този текстов файл съдържа изходния код на програмата на C, който програмистът пише с помощта на текстов редактор. След това компилаторът за C превежда този изходен код в машинен код, който може да бъде изпълнен от компютъра. Компилаторът създава изпълним файл с разширение ".exe"

3. В какъв тип файлове се съхраняват програмите на C?

-Етапите са следните:

1. Написване на код.
2. Компиляция.
3. Препроцесиране
4. Линкване (при необходимост).
5. Изпълнение на програмата.
6. Извеждане на резултатите.

4. В кой етап от жизнения цикъл на една програма се включват библиотеките?

-Библиотеките се включват в етапа на линкване на програмата(етапа след компилитането).

5. В кой етап се обработват директивите на препроцесора?

-Директивите на препроцесора в програмите на C се обработват в етапа на препроцесиране. Този етап предшества компилацията и се използва за преобразуване на изходния код преди компилаторът да го обработи.

6. В кой етап сорс кода се превръща в обектен код?

-Изходният код (source code) на програмата се превръща в обектен код по време на етапа на компилация. В този процес компилаторът преобразува изходния код, написан на програмния език, в обектен код, който е форма на машинен код, разбираема от компютъра, но още не е окончателният изпълним код.

7. В кой етап приложението взаимодейства с потребителя?

- Приложението взаимодейства с потребителя в етапа на изпълнение (runtime). В този момент програмата е стартирана и работи, а потребителят може да взаимодейства с нея чрез въвеждане на данни, използване на интерфейса на приложението и получаване на резултати от обработката на програмата. Това е фазата, в която приложението изпълнява своята функционалност и отговаря на действията на потребителя.

8. В кой етап се обработват данните?

-Данните се обработват в етапа на изпълнение (runtime) на програмата, когато програмата се стартира и реално изпълнява операции върху входните данни, след като премине през етапите на компилация и линкване.

9. За какво се използват функциите printf() и scanf()? Дайте примери.

9.1. printf(): Тази функция се използва за извеждане на данни на стандартния изход, като конзолата. Тя приема форматиран низ и няколко аргумента, които се вмъкват в низа, според форматиращите спецификатори.

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

int main()
{
    printf('Hello!')
}
```

```
C:\protokol 1>v_9.1.2.exe
Hello!
```

9.2. scanf(): Тази функция се използва за въвеждане на данни от потребителя чрез стандартния вход, като конзолата. Тя също използва форматиращи спецификатори, за да интерпретира входните данни.

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

int main()
{
    str a;
    scanf("%s",&a);
    printf("%d",a);
}
```

```
C:\protokol 1>scanf.exe
3
3
```

10. За какво се използват форматните спецификатори при извикване на функциите printf() и scanf()? Дайте примери!

- Форматните спецификатори при извикването на функциите printf() и scanf() се използват за дефиниране на формата на данните, които програмата трябва да изведе или въведе. Те определят как числата, символите и други данни трябва да бъдат форматиращи и интерпретирани. Примери %d за int, %c за char и %f за float.

11. Как се поставят и за какво се използват коментари в програма на C? Дайте пример за едноредов и многоредов комен.

Едноредов коментар се поставя с две наклонени черти (//) и важи до края на реда, а многоредов коментар се поставя между /* и */ и може да обхване множество редове.

```
#include <stdio.h>
void main()
{
    int a;
    scanf("%d",&a);
    printf("%d",a);
    // 1 line comennt
    /* 3
       line
       comment
    */
}
```

12. Може ли да съществуват вложени коментари в програма на езика C?

- В езика C, вложените коментари не са допустими. Това означава, че не можете да поставяте многоредов коментар `/* ... */` в друг многоредов коментар. Ако се опитате, кодът може да се поведе неочаквано и да предизвика компилационни грешки.

13. . Какво представлява стандартната библиотека в езика C?

- Стандартната библиотека на C включва функции за работа с вход и изход, работа със символни низове, работа с файлове, алокация на памет, математически операции, управление на символи и други. Тя предоставя също така важни типове данни като цели числа, числа с плаваща запетая, указатели и структури.

14. Какво е необходимо да направим за да използваме функции от стандартната библиотека на C?

- Включете заглавните файлове: За да имате достъп до функциите и типовете данни от стандартната библиотека, трябва да включите съответните заглавни файлове във вашия изходен код. Например, ако искате да използвате функции за вход и изход от стандартната библиотека, включете заглавния файл `<stdio.h>`. Използвайте директивата `#include` за това

15. Опишете процесите на компилиране, свързване и изпълнение на приложение.

1.Компилиране:

- Компиляцията е първата стъпка при създаването на програма.
- Компиляторът (като например GCC за C) превежда изходния код от програмния език в машинен код.
- Процесът на компилация създава обектни файлове, които съдържат машинен код, но още не са готови за изпълнение.

2.Свързване (линкване):

- След компилацията, ако програмата използва външни библиотеки или модули, те трябва да бъдат свързани с обектния код, за да се създаде изпълним файл.
- Линкерът (като например LD) извършва този процес на свързване, което включва идентифициране и резолюция на символи, които се използват от различните части на програмата.

3.Изпълнение:

- След като е бил създаден изпълним файл, този файл може да бъде стартиран на компютъра.
- Програмата се зарежда в паметта и се изпълнява от операционната система.
- Приложението работи и обработва данни според своята цел.

16. Какво е променлива в контекста на програмирането?

-В контекста на програмирането, променливата представлява именуван контейнер, който съдържа стойност или данни, които могат да се променят по време на изпълнението на програмата.

17Какво не е наред с всяко от изброените имена на променливи?


- `_name_` (не може да започва с `_`)
- `sum 1` (не може да има space)

- 3variable (не може да започва с число)
- temp-sum-1 (не може да има тире в името)
- \$dollar (не може да започва със символи)
- num_0043-(не може да има тире в името)

2)Задачи


1 и 2.

```
C:\protokol 1\zadachi>cl /c /W3 zad3.c
```

 zad3

- Компилираме и в папката при файла се създава .obj файла

```
C:\protokol 1\zadachi>link zad3.obj /OUT:zad3.exe
```

 zad3

- Линкуваме и се появява .exe файла

```
C:\protokol 1\zadachi>zad3.exe
```

- Стартираме приложението

3.

```
#include <stdio.h>
#include <string.h>
#include <locale.h>
void main()
{
    setlocale(LC_ALL, "en_US.UTF-8");
    printf("Hello world!\n");
    printf("Здравей свят!\n");
    printf("Hello\nworld\n");
}
```

```
C:\protokol 1\zadachi>zad3.exe
Hello world!
??????? ????!
Hello
world
```

4.

```
C:\protokol 1\zadachi>zad4.exe
Georgi
Pektov
Hello, Georgi Pektov
```

```
#include <stdio.h>
void main()
{
    char f_name[10], l_name[10];
    scanf("%s", &f_name);
    scanf("%s", &l_name);
    printf("Hello, %s %s\n", f_name, l_name);
}
```

5.

```
Ivan
Ivanov
23
Hello, Ivan Ivanov.
Your age is 23
```

```
#include <stdio.h>
void main()
{
    char f_name[10], l_name[10];
    int age;
    scanf("%s", &f_name);
    scanf("%s", &l_name);
    scanf("%d", &age);
    printf("Hello, %s %s.\nYour age is %d\n", f_name, l_name, age);
}
```

6.

```
#include <stdio.h>
void main()
{
    char first_p[] = "He", second_p[] = "llo", third_p[] = "wo", last_p[] = "rld";
    printf("%s%s %s%s", first_p, second_p, third_p, last_p);
}
```

```
C:\protokol 1\zadachi>zad6.exe
Hello world
```



```
#include <stdio.h>
void main()
{
    char first_p[] = "He", second_p[] = "llo", third_p[] = "wo", last_p[] = "rld";

    printf("%s%s \n %s%s",first_p, second_p, third_p,last_p);
}
```

```
Hello
world
```

7.

```
#include <stdio.h>
void main()
{
    int a;
    scanf("%d",&a)
    printf("Your number is: %d")
}
```

```
C:\protokol 1\zadachi>zad7.exe
3
Your number is: 3
```

8.

```
Your number is: 0.000000
```

```
#include <stdio.h>
void main()
{
    int a=5;
    printf("Your number is: %f",a);
}
```

-На конзолата се показва 0.0000 каквото и число да присвоим на а, това е, защото %f е се отнася за float, който не е целочислен тип

9.

```
#include <stdio.h>
void main()
{
    int a=5;
    printf("Your number is: %c",a);
}
```

Your number is: 5

-Конзолата се бърка и печата този символ, защото тя очаква след спецификатора %c да и бъде подадена променлива от тип char, а ние и подаваме integer

10.

```
#include <stdio.h>
void main()
{
    float a;
    scanf("%f", &a);
    printf("Your number is: %f",a);
}
```

3.1415
Your number is: 3.141500

11.

```
#include <stdio.h>
void main()
{
    float a=4.22123;
    printf("Your number is: %d",a);
}
```

Your number is: 536870912

12.

Your number is:

```
#include <stdio.h>
void main()
{
    float a=4.22123;
    printf("Your number is: %c",a);
}
```

13.

Type you initial here:D
Your initial is: D

```
#include <stdio.h>
void main()
{
    char your_initial;
    printf("Type you initial here:");
    scanf("%c",&your_initial);
    printf("Your initial is: %c",your_initial);
}
```

14.

Your initial is: 0

```
#include <stdio.h>
void main()
{
    char your_initial="V";
    printf("Your initial is: %d",your_initial);
}
```

15.

Type you initial here:D
Your initial is: 36893488147419660288.000000

```
#include <stdio.h>
void main()
{
    char your_initial;
    printf("Type you initial here:");
    scanf("%c",&your_initial);
    printf("Your initial is: %f",your_initial);
}
```

16.

```
Int input:1
Float input:2123.23123123
Char input:Your int: 1
Your float: 2123.231201
Your char:

Second int input:222
Second float input:12323.234454
Second char input:Your int: 222
Your float: 12323.234375
Your char:
```

```
#include <stdio.h>
void main()
{
    int i;
    float f;
    char c;

    printf("Int input:");
    scanf("%d",&i);
    printf("\n");
    printf("Float input:");
    scanf("%f",&f);
    printf("\n");
    printf("Char input:");
    scanf("%c",&c);

    printf("Your int: %d\n", i);
    printf("Your float: %f\n", f);
    printf("Your char: %c\n", c);

    printf("Second int input:");
    scanf("%d", &i);
    printf("\n");
    printf("Second float input:");
    scanf("%f", &f);
    printf("\n");
    printf("Second char input:");
    scanf("%c", &c);
    printf("Your int: %d\n", i);
    printf("Your float: %f\n", f);
    printf("Your char: %c", c);
}
```

17.

```
Int input:1

Float input:2323.33

Char input:Your int: 11539505
Your float: 0.000000
Your char:  
Second int input:12312

Second float input:
Second char input:333.213
Your int: 1178624000
Your float: 0.000000
Your char: M
```

```
#include <stdio.h>
void main()
{
    int i;
    float f;
    char c;

    printf("Int input:");
    scanf("%c",&i);
    printf("\n");
    printf("Float input:");
    scanf("%d",&f);
    printf("\n");
    printf("Char input:");
    scanf("%f",&c);

    printf("Your int: %d\n", i);
    printf("Your float: %f\n", f);
    printf("Your char: %c\n", c);

    printf("Second int input:");
    scanf("%f", &i);
    printf("\n");
    printf("Second float input:");
    scanf("%c", &f);
    printf("\n");
    printf("Second char input:");
    scanf("%d", &c);
    printf("Your int: %d\n", i);
    printf("Your float: %f\n", f);
    printf("Your char: %c", c);
}
```

18.

```
Int input:dcf
Float input:
Char input:Your int: 13309105Your float: 0.000000Your char: dSecond int input:
Second float input:
Second char input:Your int: 13309105Your float: 0.000000Your char: c
```

```
#include <stdio.h>
void main()
{
    int i;
    float f;
    char c;

    printf("Int input:");
    scanf("%d",&i);
    printf("\n");
    printf("Float input:");
    scanf("%f",&f);
    printf("\n");
    printf("Char input:");
    scanf("%c",&c);

    printf("Your int: %d", i);
    printf("Your float: %f", f);
    printf("Your char: %c", c);

    printf("Second int input:");
    scanf("%d", &i);
    printf("\n");
    printf("Second float input:");
    scanf("%f", &f);
    printf("\n");
    printf("Second char input:");
    scanf("%c", &c);
    printf("Your int: %d", i);
    printf("Your float: %f", f);
    printf("Your char: %c", c);
}
```

19.

```
Int input-Float input-Char input:1 2.2 f
Your int: 1
Your float: 2.200000
Your char: f
Second int input-Second float input-Second char input:3 23.3232 g
Your int: 3
Your float: 23.323200
Your char: g
```

```
#include <stdio.h>
void main()
{
    int i;
    float f;
    char c;

    printf("Int input-");
    printf("Float input-");
    printf("Char input:");
    scanf("%d %f %c", &i, &f, &c);
    printf("\n");
    printf("Your int: %d", i);
    printf("\n");
    printf("Your float: %f", f);
    printf("\n");
    printf("Your char: %c", c);
    printf("\n");
    printf("Second int input-");
    printf("Second float input-");
    printf("Second char input:");
    scanf("%d %f %c", &i, &f, &c);
    printf("\n");
    printf("Your int: %d", i);
    printf("\n");
    printf("Your float: %f", f);
    printf("\n");
    printf("Your char: %c", c);
}
```