

Napište program, ktorý zo štandardného vstupu načíta reálne číslo x . Do súboru `nasobky.txt` zapíše 1, 2, ..., 10- násobky čísla x . Súbor má obsahovať 10 riadkov s nasledujúcim formátovaním: v i -tom riadku vypíše $i * x = ix$, kde i je číslo riadku na 2 miesta, x je načítané číslo vypísané na 2 desatinné miesta a ix je i -ty násobok čísla x tiež vypísaný na 2 desatinné miesta. Každý riadok je ukončený znakom konca riadku.

Ukážka vstupu:

2.5

Súbor `nasobky.txt` pre ukážkový vstup:

```
1 * 2.50 = 2.50
2 * 2.50 = 5.00
3 * 2.50 = 7.50
4 * 2.50 = 10.00
5 * 2.50 = 12.50
6 * 2.50 = 15.00
7 * 2.50 = 17.50
8 * 2.50 = 20.00
9 * 2.50 = 22.50
10 * 2.50 = 25.00
```

```
1 #define _CRT_SECURE_NO_WARNINGS
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 int main() {
6     FILE* fp;
7     int i;
8     fp = fopen("nasobky.txt", "w+");
9     double cislo, nasobok;
10    scanf("%lf",&cislo);
11    for (i = 1; i <= 10; i++)
12    {
13        nasobok = cislo * i;
14        fprintf(fp, "%d %s %.2f %s %.2f\n",i, "*",cislo,"=",nasobok);
15    }
16    fclose(fp);
17
18
19    return(0);
20 }
```

Napište program, ktorý zo štandardného vstupu (klávesnice) načíta znak. Ďalej číta znaky zo súboru znak.txt. Ak program prečítal zo štandardného vstupu (klávesnice) 's', vypisuje načítané znaky do súboru novy.txt. Ak načítal ľubovoľný iný znak, vypisuje načítané znaky na štandardný výstup (obrazovku). Súbor novy.txt alebo štandardný výstup bude teda obsahovať presnú kópiu obsahu súboru znak.txt.

Ukážka vstupu:

s

Ukážka súboru znak.txt:

abracadabra
bubu

Súbor novy.txt pre ukážkový vstup:

abracadabra
bubu

uloha5-2.c

novy.txt

znak.txt

```

1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  int main() {
6      FILE *fr ,*fw;
7      int i;
8      char vstup, c, durhyznak;
9
10     fr = fopen("znak.txt", "r+");
11     fw = fopen("novy.txt", "w");
12     double cislo, nasobok;
13
14     scanf("%c", &vstup);
15
16
17
18     while ((c = getc(fr)) != EOF)
19     {
20         if (vstup == 's') {
21
22
23
24             putc(c, fw);
25         }
26         else    putchar(c);
27     }
28
29
30
31
32
33     fclose(fw);
34     fclose(fr);
35
36
37

```

Kompilácia

Štandardný vstup

1 s

Štandardný výstup

1

Napište program, ktorý číta znaky zo súboru vstup.txt po riadkoch. Každý riadok prepíše do súboru cisl1a.txt. Po každom prepísanom riadku na ďalšom riadku uvedie počet malých písmen z prečítaného riadku. Ak súbor už predtým existoval a obsahoval nejaké dáta, program tieto dáta nezmaže a svoj výstup napíše na koniec súboru cisl1a.txt. Program nečíta žiaden vstup zo štandardného vstupu a nevypisuje žiaden výstup na štandardný výstup. Predpokladajte, že posledný riadok je vždy ukončený koncom riadku.

Ukážka súboru vstup.txt:

```
ahoj123
x*Y*z
```

Ukážka súboru cisl1a.txt pred spustením programu:

```
qwerty
6
```

Súbor cisl1a.txt po spustení programu:

```
qwerty
6
ahoj123
4
x*Y*z
2
```

```
1 // uloha5-3.c -- Peter Plevko, 23.10.2019 09:31
2 #define _CRT_SECURE_NO_WARNINGS
3 #include <stdio.h>
4 #include <stdlib.h>
5 #include <string.h>
6 int main()
7 {
8
9     char str[20];
10    int countL = 0;
11    int countU;
12    int counter;
13    int c;
14    FILE* Fread;
15    FILE* Fwrite;
16
17    Fread = fopen("vstup.txt", "r");
18
19    Fwrite = fopen("cisl1a.txt", "a+");
20
21
22
23    while (fgets(str,20,Fread) != NULL)
24    {
25
26        for (counter = 0; str[counter] != NULL; counter++) {
27            if (str[counter] >= 'a' && str[counter] <= 'z')
28                countL++;
29
30        }
31
32        fprintf(Fwrite, "%s", str);
33        fprintf(Fwrite, "%d", countL);
34        fprintf(Fwrite, "\n");
35
36
37
```

Kompilácia

1

Štandardný výstup

1

Napište program, ktorý bude čítať znaky zo súboru text.txt pokiaľ nenačíta znak '*'. Ak načíta znak 'x' alebo 'X' vypíše Precital som X, ak znak 'y' alebo 'Y' vypíše Precital som Y, ak načíta znaky '#', '\$' alebo '&' vypíše Precital som riadiaci znak a ak načíta znak '*' vypíše koniec a skončí čítanie súboru. Po prečítaní súboru vypíše správu Pocet precitanych medzier: nasledovanú medzerou a počtom prečítaných medzier. Každá správa je nasledovaná koncom riadku.

Ukážka súboru text.txt:

```
$ abc 5 xyz #  
& Q *# abf
```

Ukážkový výstup pre súbor text.txt:

```
Precital som riadiaci znak  
Precital som X  
Precital som Y  
Precital som riadiaci znak  
Precital som riadiaci znak  
Koniec  
Pocet precitanych medzier: 6
```

```
1 // uloha5-4.c -- Peter Plevko, 6.11.2019 19:01  
2  
3 #define _CRT_SECURE_NO_WARNINGS  
4 #include <stdio.h>  
5 #include <stdlib.h>  
6 #include <string.h>  
7 int main()  
8 {  
9  
10     char str[20];  
11     int countL = 0;  
12     int countU;  
13     int counter;  
14     int s=0;  
15     FILE* Fread;  
16     char c;  
17  
18     Fread = fopen("text.txt", "r");  
19  
20  
21  
22  
23  
24     while ((c = fgetc(Fread)) != EOF)  
25     {  
26         if (c == ' ')s++ ;  
27         if (c == '#' || c == '$' || c == '&')printf("Precital som riadiaci znak\n");  
28         if (c == 'x' || c == 'X') printf("Precital som X\n");  
29         if (c == 'y' || c == 'Y') printf("Precital som Y\n");  
30         if (c == '*')  
31         {  
32             printf("Koniec\n");  
33             printf("Pocet precitanych medzier: %d", s);  
34             return 0;  
35         }  
36     }  
37
```

Kompilácia

1

Štandardný výstup

1

Napište program, ktorý určí, či majú dva súbory prvý.txt a druhý.txt rovnaký obsah. Program nečíta žiadne dáta zo štandardného vstupu. Ak majú súbory rovnaký obsah, program vypíše Subory su identicke Ak súbory rovnaký obsah nemajú, vypíše program Pocet roznych znakov: nasledovaný medzerou, počtom rôznych znakov v súboroch a ukončený koncom riadku. i-ty znak v jednom súbore považujte za rôzny od i-teho znaku v druhom súbore, ak oba znaky existujú (t.j. ani jeden súbor nemá menej ako i znakov) a príslušné znaky sa nerovnajú. Ak majú súbory nerovnakú dĺžku, na výstup program vypíše ešte jeden riadok obsahujúci správu Jeden zo suborov je dlhší o x znakov Pričom x je počet znakov o ktoré je jeden zo súborov dlhší. Správa je nasledovaná koncom riadku.

Ukážka súboru prvý.txt:

ahoj

Ukážka súboru druhý.txt:

ahujx

*

Výstup pre ukážkové súbory:

Pocet roznych znakov: 1

Jeden zo suborov je dlhší o 3 znakov

uloha5-5.c

prvy.txt

druhy.txt

1

#define _CRT_SECURE_NO_WARNINGS

2

#include <stdio.h>

3

#include <stdlib.h>

4

#include <string.h>

5

int main()

6

{

7

8

char str[20];

9

int countL = 0;

10

int countU;

11

int counter;

12

int s=0;

13

FILE* Fread1;

14

FILE* Fread2;

15

char c,p;

16

int dlzka1=0, dlzka2=0;

17

int pocetroznychznakov=0;

18

19

Fread1 = fopen("prvy.txt", "r");

20

Fread2 = fopen("druhy.txt", "r");

21

22

23

int vysledok = 0;

24

25

while (c = fgetc(Fread1) != EOF)

26

{

27

|

28

dlzka1++;

29

}

30

31

while (p = fgetc(Fread2) != EOF)

32

{

33

| dlzka2++;

34

}

35

rewind(Fread1);

36

rewind(Fread2);

37

while (((c = fgetc(Fread1)) != EOF) && (p = fgetc(Fread2)) != EOF)

Kompilácia

Štandardný vstup

1

Štandardný výstup

1

1 1001 1287

Vytvorte funkciu, ktorá prevedie zlomok do základného tvaru. Funkcia má ako argumenty dve čísla (volané odkazom). Argumenty vyjadrujú čitateľa a menovateľa zlomku. Funkciu otestujte v krátkom programe, ktorého vstupom je jeden riadok obsahujúci 2 celé čísla oddelené medzerou. Prvé z čísel predstavuje čitateľa a druhé menovateľa zlomku. Výstupom programu je riadok obsahujúci správu Zakladny tvar zlomku: c/m , kde c je čitateľ a m menovateľ zlomku v základnom tvare.

Ukážka vstupu:

12 60

Výstup pre ukážkový vstup:

Zakladny tvar zlomku: 1/5

```
// uloha5-6.c -- Peter Plevko, 10.11.2019 12:05

#include <stdio.h>

void uprav_zlomok( int * citatel, int * menovatel)
{
    int i;
    for (i = 2; i <= *citatel; i++)
        while ( *citatel % i == 0 && *menovatel % i == 0 )
        {
            *menovatel/=i;
            *citatel/=i;
        }
}

int main()
{
    int x, y;
    scanf("%d %d", &x, &y);
    uprav_zlomok( &x,&y );
    printf("Zakladny tvar zlomku: %d/%d\n", x, y);
    return 0;
}
```

1

Napište program na výpočet obsahu a obvodu obdĺžnika. Vytvorte funkciu `nacitaj`, ktorá načíta a pomocou argumentov vráti dve reálne čísla. Ďalej vytvorte funkciu `vypocitaj`, ktorá ako argumenty dostane dĺžky strán obdĺžnika a prostredníctvom argumentov vráti obsah a obvod obdĺžnika. V hlavnom programe volajte funkciu `nacitaj` na načítanie rozmerov obdĺžnika a funkciu `vypocet` na výpočet obsahu a obvodu obdĺžnika. Program vypíše obsah a obvod obdĺžnika zaokrúhlený na tri desatinné miesta.

Ukážka vstupu:

3.5 4.75

Výstup pre ukážkový vstup:

Obsah: 16.625

Obvod: 16.500

```
// uloha5-7.c -- Peter Plevko, 10.11.2019 12:05

#include <stdio.h>

int nacitaj( double *s1,double *s2 )
{
    return scanf( "%lf %lf",s1,s2 ) == 2 ;
}

void vypocitaj(double x, double y, double *obsah,double*obvod )
{
    *obsah = x*y;
    *obvod = 2*(x+y) ;
}

int main()
{
    double x, y, obsah, obvod;
    if (!nacitaj( &x,&y ))
    {
        printf("Zly vstup\n");
        return 0;
    }
    vypocitaj( x,y,&obsah,&obvod );
    printf("Obsah: %.3lf \n", obsah);
    printf("Obvod: %.3lf \n", obvod);
    return 0;
}
```

1 33.3 47.7

1

Napište program, ktorý rekurzívnou funkciou vypočíta súčet nepárnych kladných celých čísel menších ako N. Na vstupe je dané kladné celé číslo N, na výstup program vypíše jedno číslo: súčet všetkých nepárnych kladných celých čísel menších ako N.

Ukážka vstupu:

10

Výstup pre ukážkový vstup:

25

```
// uloha5-8.c -- Peter Plevko, 10.11.2019 12:05
```

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

```
int neparne(int n)
```

```
{
```

```
    if ( n<1 )
```

```
        return 0;
```

```
    if ( n%2!=0 )
```

```
        return n+neparne(n-1) ;
```

```
    return neparne(n-1);
```

```
}
```

```
int main()
```

```
{
```

```
    int n;
```

```
    scanf("%d", &n);
```

```
    printf("%d\n", neparne(n-1) );
```

```
    return 0;
```

```
}
```

```
11
12
13
14
15
16
17
```

1 45

1

Napište program, ktorý rekurzívnou funkciou vypočíta ciferný súčet čísla. Na vstupe je dané kladné celé číslo N, na výstup program vypíše jedno číslo: ciferný súčet čísla N.

Ukážka vstupu:

56

Výstup pre ukážkový vstup:

11

// uloha5-9.c -- Peter Plevko, 10.11.2019 12:05

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

```
int cifsucet(int x)
```

```
{
```

```
    if ( x<1 )
```

```
        return 0;
```

```
    return x%10+cifsucet(x/10) ;
```

```
}
```

```
int main()
```

```
{
```

```
    int n;
```

```
    scanf("%d", &n);
```

```
    printf("%d\n", cifsucet(n));
```

```
    return 0;
```

```
}
```

11
11
11
11
11
11

1