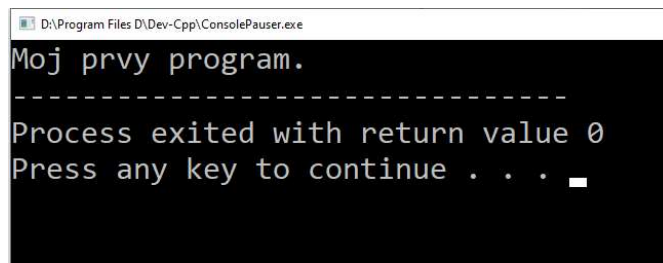
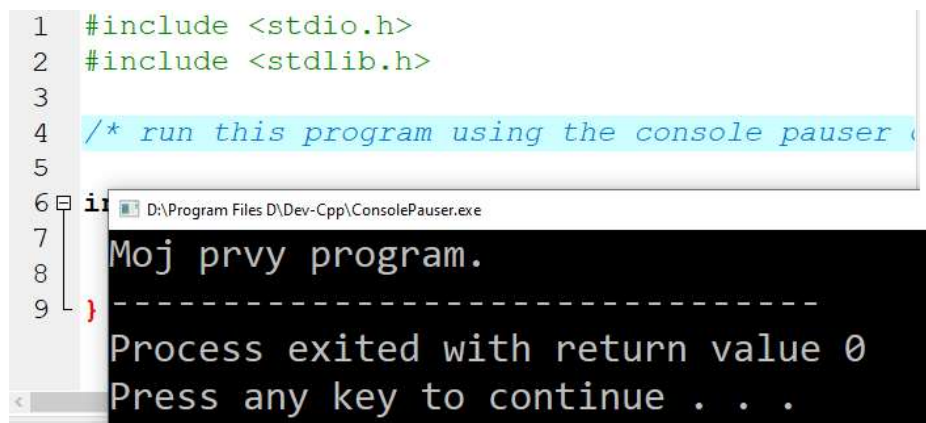


Úloha 1: Inštalácia prostredia [Dev-C++](https://dev-cpp.com/) (<https://dev-cpp.com/>)

Nainštalujte si prostredie v ktorom môžete testovať vaše programy (Dev-C++, Visual Studio, CLion, Code:Blocks) a vytvorte si vlastný program v programovacom jazyku C, ktorý vypíše na obrazovku vetu "Moj prvý program."



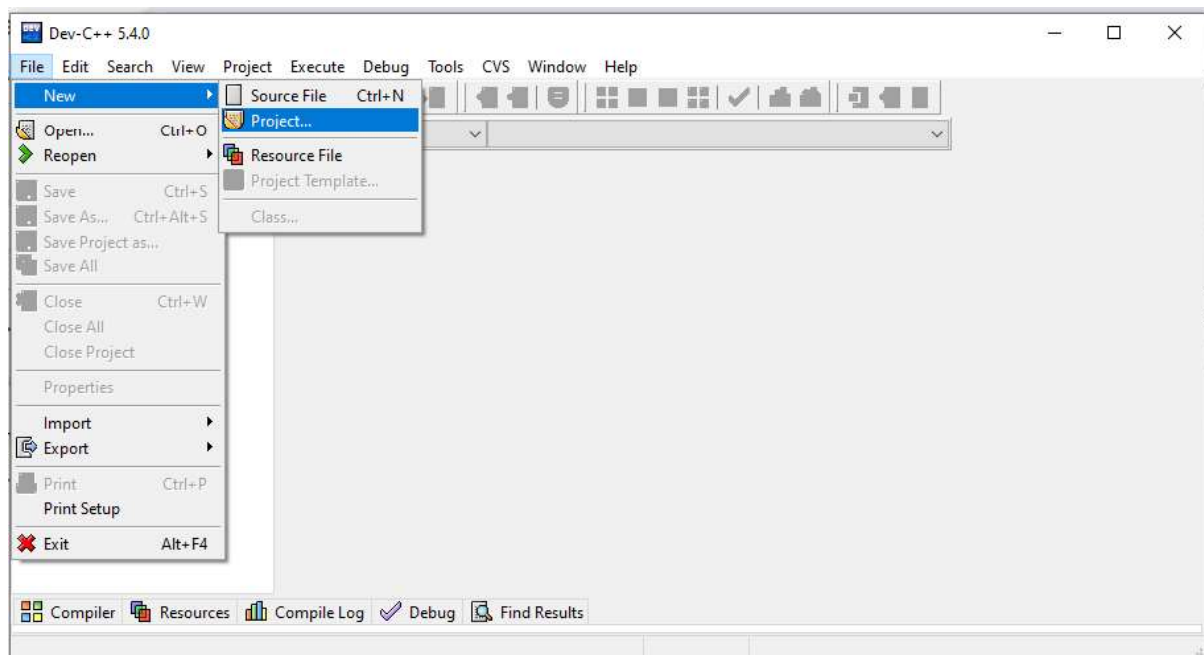
Správne riešenie:



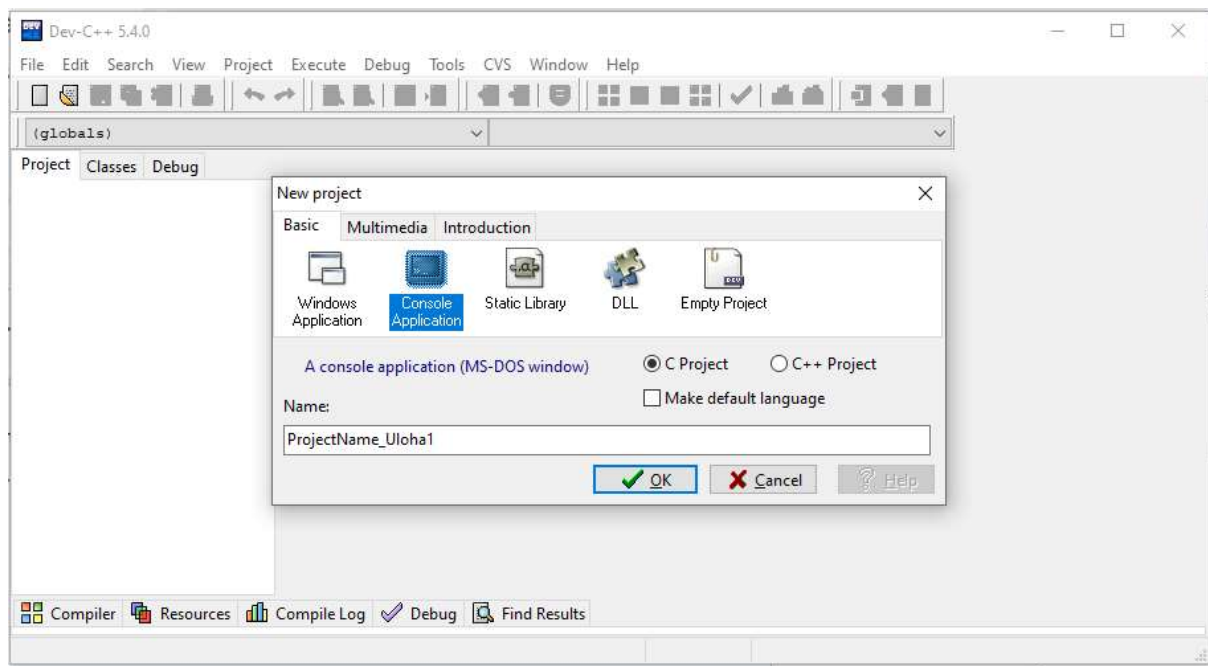
Nápoved':

Krok 1: inštalácia: <https://dev-cpp.com/>

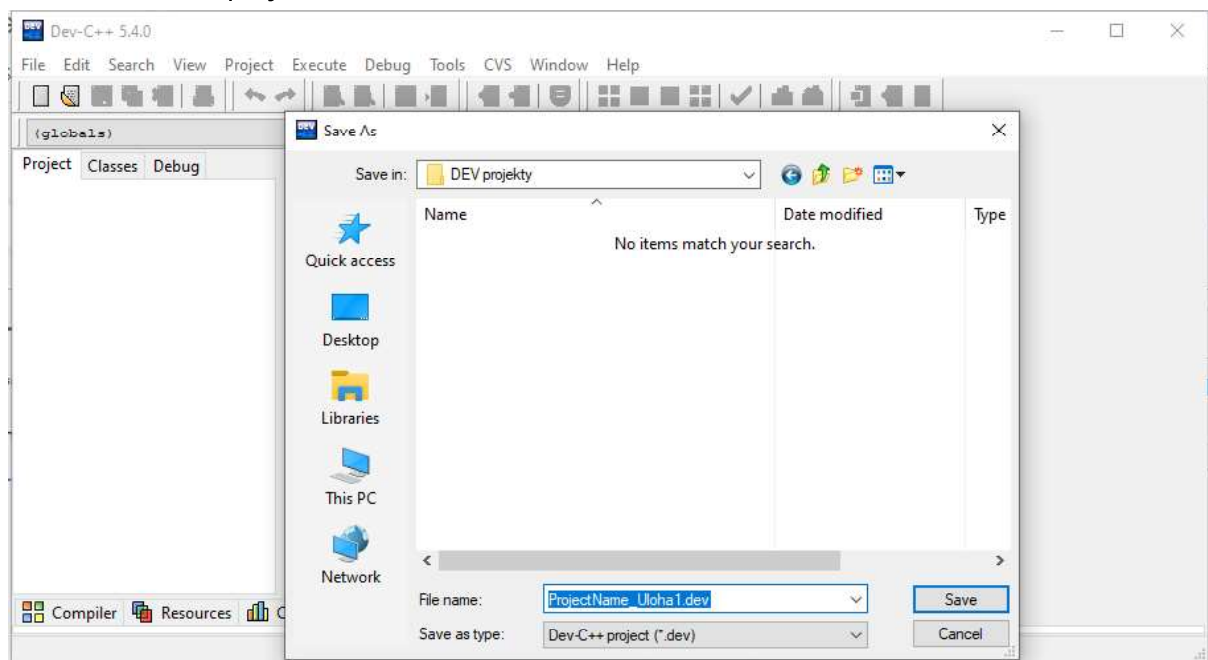
Krok 2: File -> New -> Project...



Krok 3: Vytvoríme si konzolovú aplikáciu pre C projekt a vložíme meno projektu -> OK

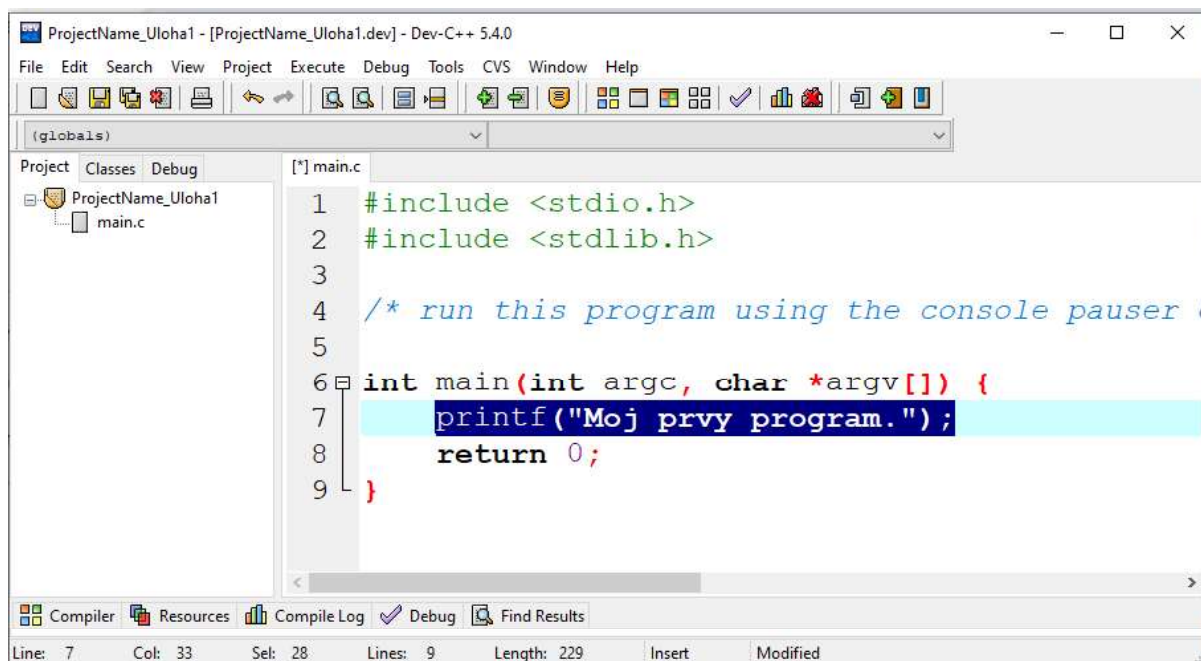


Krok 4: Uložíme projekt na vhodné miesto na disku. -> Save

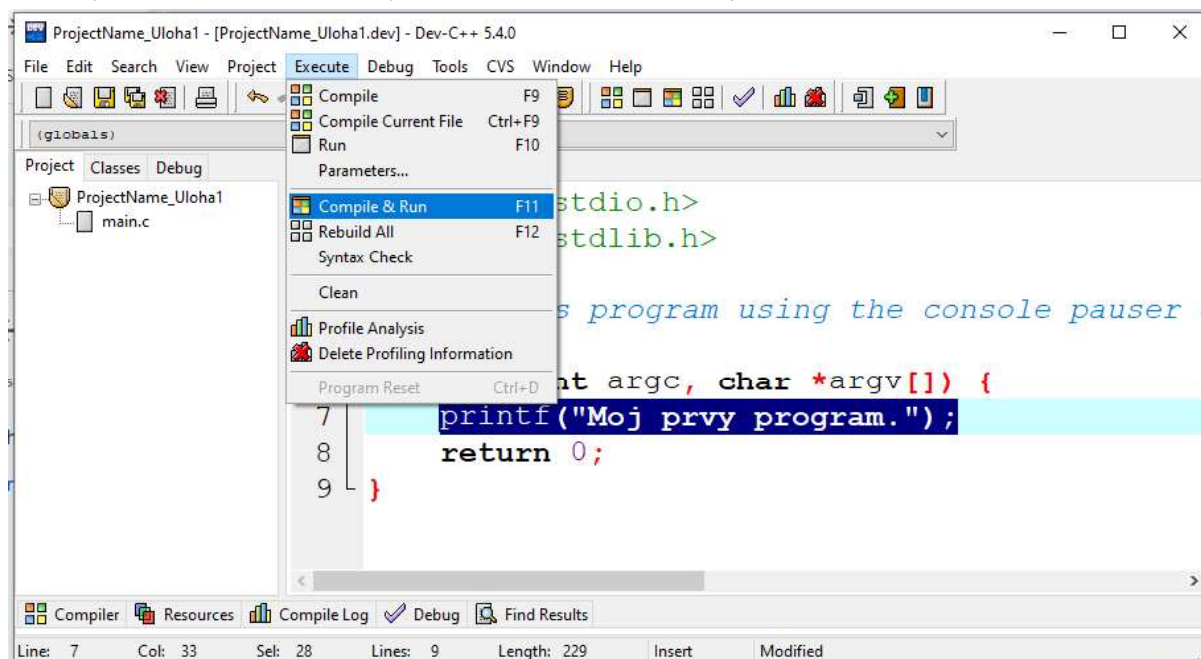


Krok 5: Upravíme predpripravený program nasledovne.

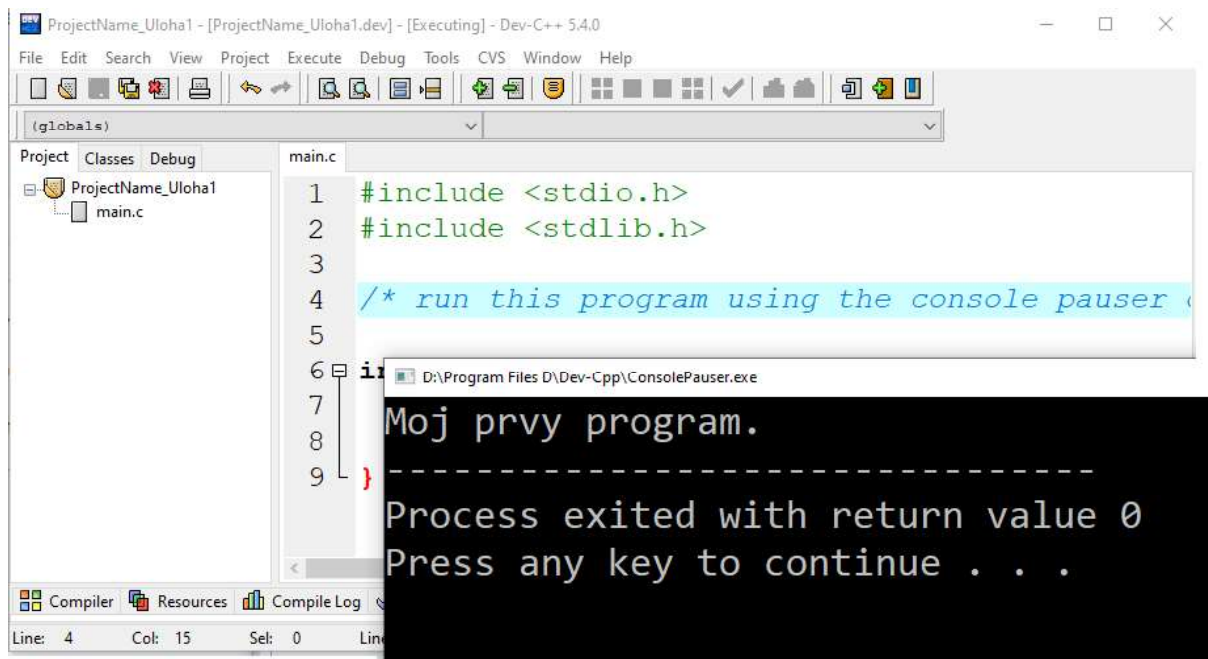
```
printf("Moj prvý program.");
```



Krok 6: Prostredníctvom F11 program skompilujeme a spustíme. Ak zdrojový súbor nebol uložený, prostredie sa nás opýta na uloženie mena zdrojového súboru.



Krok 7: Program sa spustí a vykoná jednotlivé príkazy v programe.



Časová náročnosť: 30min.

Úloha 2: Inštalácia gcc a spustenie programu

Nainštalujte si GNU compiler a prostredníctvom gcc kompilera spustíte svoj program. Program vypíše vetu "Program funguje." <http://sourceforge.net/projects/mingw/>

Správne riešenie:

```
C:\Users\JanoPC_1\Documents\C programs>run
Program funguje.
C:\Users\JanoPC_1\Documents\C programs>_
```

Nápoved':

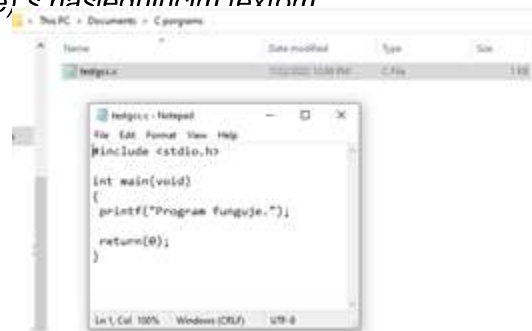
Odporúčaný manuál pre inštaláciu Youtube video:

https://www.youtube.com/watch?v=8lb7nwc33uA&ab_channel=DerekBanas

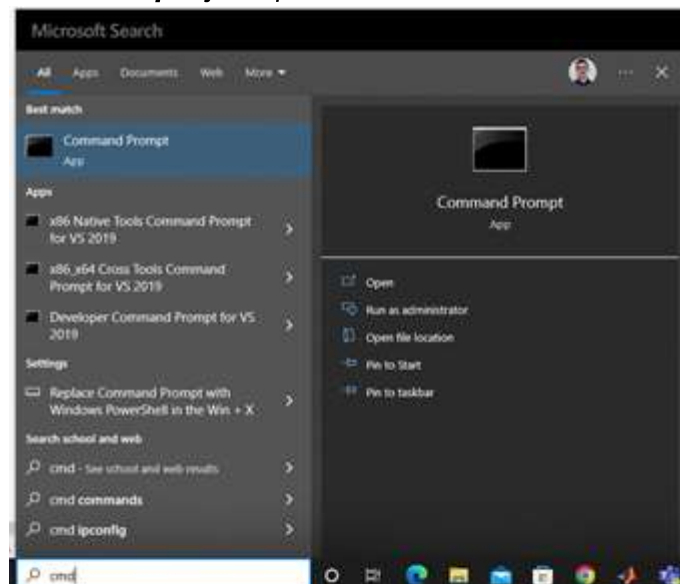
Krok 1: Inštalácia podľa manuálu

Krok 2: Vytvorte si program (napr. aj v Notepad-e) s nasledujúcim textom

```
#include <stdio.h>
int main(void)
{
    printf("Program funguje.");
    return(0);
}
```



Krok 3: Otvor **Command Prompt** aj cez príkaz **cmd**



Krok 4: Nastavte sa na adresár, kde máme uložený Váš zdrojový súbor (v ukážke je na adrese C:\Users\JanoPC_1\Documents\C programs)

```
C:\> Command Prompt

Microsoft Windows [Version 10.0.19043.1826]
(c) Microsoft Corporation. All rights reserved.

C:\Users\JanoPC_1>cd C:\Users\JanoPC_1\Documents\C programs

C:\Users\JanoPC_1\Documents\C programs>
```

Krok 5: Z Vášho zdrojového kódu vytvoríme spustiteľný kód prostredníctvom prepínača `-o`

```
C:\Users\JanoPC_1\Documents\C programs>
C:\Users\JanoPC_1\Documents\C programs>gcc testgcc.c -o run
```

Krok 6: Prostredníctvom spustiteľného súboru `run.exe` spustíme náš program.

```
C:\Users\JanoPC_1\Documents\C programs>run
Program funguje.
C:\Users\JanoPC_1\Documents\C programs>_
```

Časová náročnosť: 20 min.

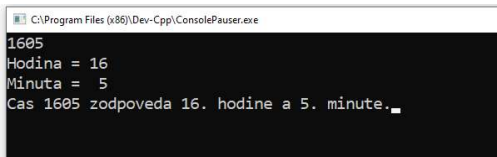
Úloha 3: Štandardný vstup/výstup, práca s celými číslami

Zo štandardného vstupu (klávesnice) načítajte celé 4-ciferné číslo do premennej typu `int` s názvom `cas`. Zadefinujte ďalšie 2 premenné typu `int` s názvami `hodiny` a `minuty`. Do premennej `hodiny` vypočítajte hodnotu zodpovedajúcu prvým 2 cifrám (2 rády) z premennej `cas`, a do premennej `minuty` zapíšte zase posledné 2 cifry z hodnoty premennej `cas`. Následne vypíšte na obrazovku hodinu aj minútu samostatne, aj v tvare vety.

Vstup: 1605

Výstup:

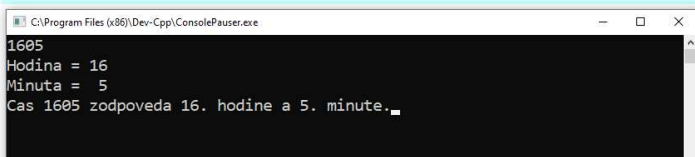
```
Hodina = 16
Minuta = 5
Cas 1605 zodpoveda 16. hodine a 5. minute.
```



```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
1605
Hodina = 16
Minuta = 5
Cas 1605 zodpoveda 16. hodine a 5. minute._
```

Správne riešenie:

```
cas.cpp
1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7      int cas;           // definovanie premenných a ich typov
8      int hodiny, minuti;
9
10     scanf("%d", &cas); // Nacitanie vstupu z klavesnice
11
12     hodiny = cas / 100; // vypocet poctu hodin; znak / je celociselné delenie
13     minuti = cas % 100; // vypocet poctu minut; znak % reprezentuje zvyšok po delení
14
15     printf("Hodina = %2d\n", hodiny); // vypisy na obrazovku
16     printf("Minuta = %2d\n", minuti); // \n reprezentuje zariadkovanie
17     printf("Cas %d zodpoveda %d. hodine a %d. minute.", cas, hodiny, minuti);
18
19     getchar(); // sluzi na zastavenie na konci programu
20     getchar(); // niektore prostredia totiž automaticky uzavru okno
21     return 0;
22 }
23
```



```
C:\Program Files (x86)\Dev-Cpp\ConsolePauser.exe
1605
Hodina = 16
Minuta = 5
Cas 1605 zodpoveda 16. hodine a 5. minute._
```

Nápoveda:

- Pre zadefinovanie premennej určitého typu a názvu použite formát `názov_typu názov_premennej ;` // [\(príklad\)](#)
- pre načítanie celého čísla zo štandardného vstupu do premennej `x` typu `int` použite funkciu: `scanf(" %d", &x);` // [\(príklad\)](#)
- pre zadefinovanie hodnôt premenných `hodiny` a `minuty` použite príkaz priradenia; na získanie príslušných rádov čísla použite operácie `/` (delenia - division) a `%` (zvyšku po delení - modulo).
- Pre výpis celého čísla na obrazovku použite funkciu:

- ```
printf("Sprievodny text %d ", premenna);
```
- Pre zarovnanie výpisov čísiel pod seba, použite vo funkcii printf formát  

```
printf("Minuta = %2d", minuty);
```

Časová náročnosť 5 min.



#### Úloha 4: Štandardný vstup/výstup, práca s reálnymi číslami, math.h, define

V jazyku C vytvorte program, ktorý zo štandardného vstupu (klávesnice) načíta 1 reálne číslo typu double. Toto číslo reprezentuje plochu štvorca, resp. kruhu v  $\text{m}^2$ . Vypočítajte a vypíšte na obrazovku veľkosť strany štvorca a polomer kruhu (ktoré majú danú plochu). Vypočítajte tiež obvod štvorca a kruhu a taktiež ich vypíšte na obrazovku. Výpis na obrazovku realizujte podľa nižšie uvedeného vzoru, na 4 desatinné miesta. Program dôkladne otestujte.

Vstup: 20.5

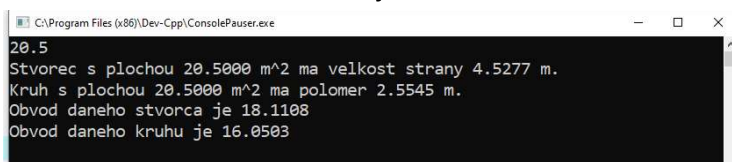
Výstup:

Stvorec s plochou 20.5000  $\text{m}^2$  ma veľkost strany 4.5277 m.

Kruh s plochou 20.5000  $\text{m}^2$  ma polomer 2.5545 m.

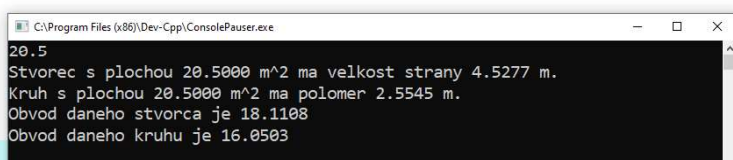
Obvod daneho stvorca je 18.1108

Obvod daneho kruhu je 16.0503



#### Správne riešenie:

```
Uloha4-geometria.cpp
1 #include <stdio.h> // kniznica stdio.h na dostupnost funkcii printf()/scanf()/getchar()
2 #include <math.h> // potrebujeme kniznicu math.h, kedze pouzivame funkcie sqrt() a pow();
3
4 #define PI (3.14159265)
5
6 int main(void)
7 {
8 // definovanie potrebných premenných typu double = realne cislo
9 double Area;
10 double a, r;
11
12 scanf("%lf", &Area); // nacitanie plochy ako realneho cisla zo vstupu (klavesnice)
13
14 // vypocet velkosti strany stvorca / polomeru kruhu
15 a = pow(Area, 0.5); // alebo tiez a = sqrt(Area);
16 r = sqrt(Area / PI);
17
18 // Vypisy na obrazovku
19 printf("Stvorec s plochou %.4lf m^2 ma veľkost strany %.4lf m. \n", Area, a);
20 printf("Kruh s plochou %.4lf m^2 ma polomer %.4lf m. \n", Area, r);
21 printf("Obvod daneho stvorca je %.4lf \n", 4.0 * a);
22 printf("Obvod daneho kruhu je %.4lf \n", 2*PI*r);
23
24 getchar();
25 getchar();
26
27 return 0;
28 }
29
```



#### Nápoveda:

- Konštantu PI zadefinujte v úvode programu nasledovne: `#define PI (3.14159265)`
- Pri riešení použite knižnicu `math.h` (pripojte ju použitím `#include<math.h>` ) z ktorej použite funkciu `sqrt(x)` na výpočet odmocniny, alebo funkciu `pow(x, 0.5)` na výpočet mocniny. [sqrt inšpirácia](#)

- Zaokrúhlenie na 4 desatinné miesta realizujte až pri formátovanom výpise v tvare `printf("%.4lf", x);`

Časová náročnosť 5 min.

### Úloha 5: Štandardný vstup/výstup, práca s číslami, zaokrúhľovanie

V jazyku C vytvorte program, ktorý zo štandardného vstupu (klávesnice) načíta 1 reálne číslo typu double. Toto číslo zaokrúhlite na 4 desatinné miesta. Výpis na obrazovku realizujte podľa nižšie uvedeného vzoru. Program dôkladne otestujte.

Vstup: 21.12357

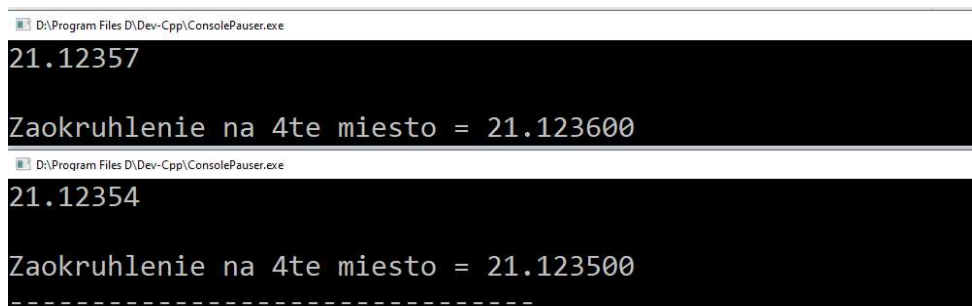
Výstup:

Zaokruhlenie na 4te miesto = 21.1236

Vstup: 21.12354

Výstup:

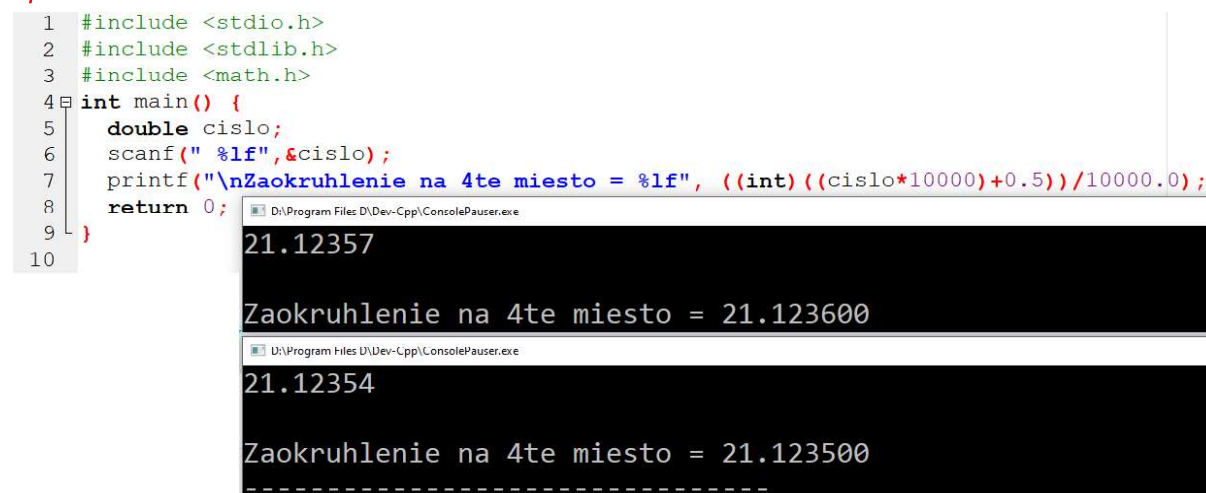
Zaokruhlenie na 4te miesto = 21.1235



```
D:\Program Files D\Dev-Cpp\ConsolePauser.exe
21.12357
Zaokruhlenie na 4te miesto = 21.123600
D:\Program Files D\Dev-Cpp\ConsolePauser.exe
21.12354
Zaokruhlenie na 4te miesto = 21.123500

```

#### Správne riešenie:



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4 int main() {
5 double cislo;
6 scanf("%lf", &cislo);
7 printf("\nZaokruhlenie na 4te miesto = %lf", ((int)((cislo*10000)+0.5))/10000.0);
8 return 0;
9 }
10
```

```
D:\Program Files D\Dev-Cpp\ConsolePauser.exe
21.12357
Zaokruhlenie na 4te miesto = 21.123600
D:\Program Files D\Dev-Cpp\ConsolePauser.exe
21.12354
Zaokruhlenie na 4te miesto = 21.123500

```

#### Nápoved':

- tj. ak 5té desatinné miesto je väčšie nanajvýš rovné číslu 5, 4 desatinné miesto zvýšiť o 1
- je možné použiť pretypovanie premenej, tj. z čísla double spravim integer, týmto sa čísla za desatinnou čiarkou strácajú

Časová náročnosť: 6 min.

### Úloha 6: Štandardný vstup/výstup, práca s znakom

V jazyku C vytvorte program, ktorý načíta znak z klávesnice a vypíše jeho dekadickú hodnotu v ASCII tabuľke a načíta celé číslo (číslo od 0 -127) z klávesnice a vypíše jeho hodnotu v ASCII tabuľke.

```
D:\Program Files D\Dev-Cpp\ConsolePauser.exe
a 97
Nacitany znak: a ma hodnotu 97
Nacitane cislo: 97 ma znak v ASCII tabulke a

```

*Správne riešenie:*

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4
5
6 int main() {
7 char znak;
8 int cislo;
9 scanf(" %c %d",&znak, &cislo);
10 printf("Nacitany znak: %c ma hodnotu %d",znak,znak);
11 printf("\nNacitane cislo: %d ma znak v ASCII tabulke %c",cislo,(char)cislo);
12 return 0;
13 }
```

```
D:\Program Files D\Dev-Cpp\ConsolePauser.exe
a 97
Nacitany znak: a ma hodnotu 97
Nacitane cislo: 97 ma znak v ASCII tabulke a

```

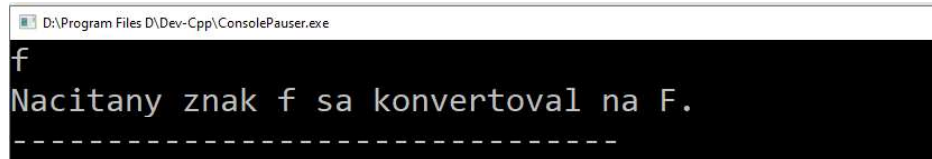
*Nápoved':*

- [ASCII tabuľka](#)

*Časová náročnosť:* 3min.

### Úloha 7: Štandardný vstup/výstup, práca s znakom

V jazyku C vytvorte program, ktorý načíta znak malého písmena z klávesnice a vypíše znak veľkého písmena. Výpis na obrazovku bude v tvare "Nacitany znak X sa konvertoval na Y."



```
D:\Program Files D\Dev-Cpp\ConsolePauser.exe
f
Nacitany znak f sa konvertoval na F.

```

*Správne riešenie:*



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main() {
4 char znak;
5 scanf(" %c",&znak);
6 printf("Nacitany znak %c sa konvertoval na %c.",znak,znak-('a'-'A'));
7 return 0;
8 }
9
10
11
12
```

```
D:\Program Files D\Dev-Cpp\ConsolePauser.exe
f
Nacitany znak f sa konvertoval na F.

```

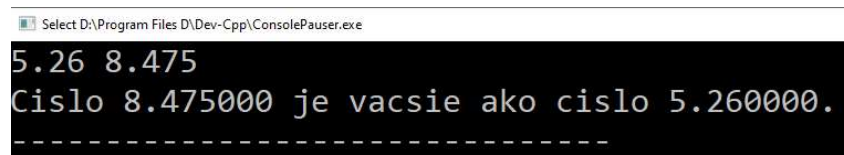
*Nápoved':*

- [ASCII tabuľka](#)
- dátový typ [char](#)

**Časová náročnosť:** 3 min.

### Úloha 8: Podmienka - jednoduchá

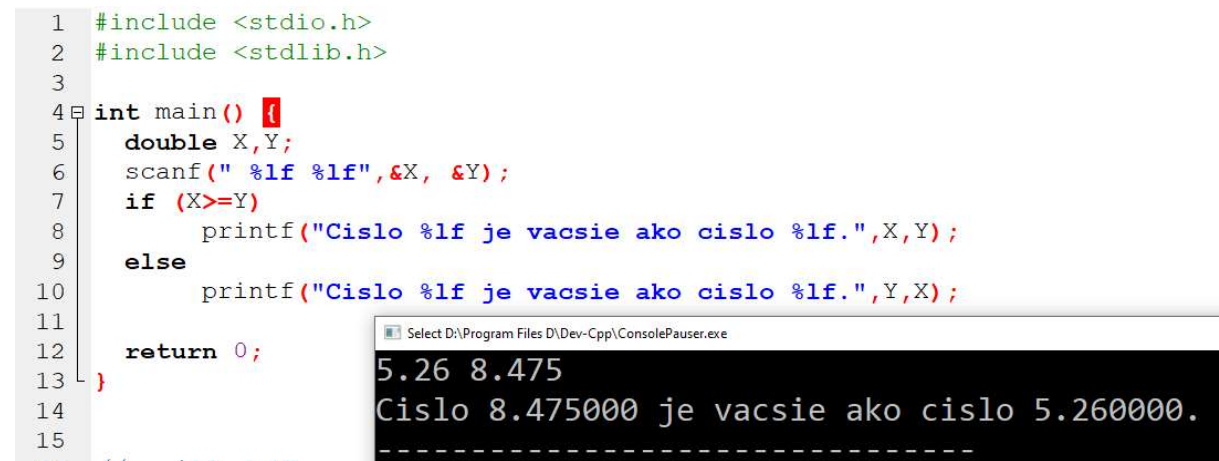
V jazyku C vytvorte program, ktorý načíta dve čísla X, Y a zisti ktoré z nich je väčšie. Výpis na obrazovku bude v tvare "Cislo X je vacsie ako cislo Y."



```
Select D:\Program Files D\Dev-Cpp\ConsolePauser.exe
5.26 8.475
Cislo 8.475000 je vacsie ako cislo 5.260000.

```

*Správne riešenie:*



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5 double X,Y;
6 scanf(" %lf %lf",&X, &Y);
7 if (X>=Y)
8 printf("Cislo %lf je vacsie ako cislo %lf.",X,Y);
9 else
10 printf("Cislo %lf je vacsie ako cislo %lf.",Y,X);
11
12 return 0;
13 }
14
15
```

```
Select D:\Program Files D\Dev-Cpp\ConsolePauser.exe
5.26 8.475
Cislo 8.475000 je vacsie ako cislo 5.260000.

```

*Nápoved':*

- v tomto prípade nemusíte riešiť problém rovnosti čísel X a Y, postačí jednoduchá podmienka `if`

*Časová náročnosť:* 5 min.

### Úloha 9: Podmienka zložitá

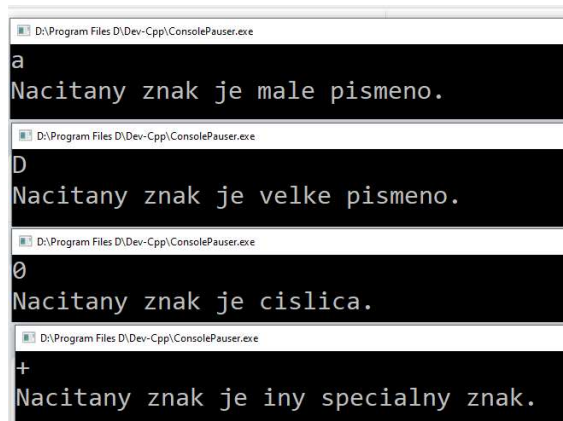
V jazyku C vytvorte program, ktorý načítá znak zo štandardného vstupu a zistí či načítaný znak je číslica (čísllice od 0 - 9), malé písmeno (znaky od a - z) alebo veľké písmeno (znaky od A- Z) alebo iný špeciálny znak. Podľa toho sa na obrazovku vypíše aj veta:

čísllica (čísllice od 0 - 9) -> 'Nacitany znak je cislica.'

malé písmeno (znaky od a - z) -> 'Nacitany znak je male pismeno.'

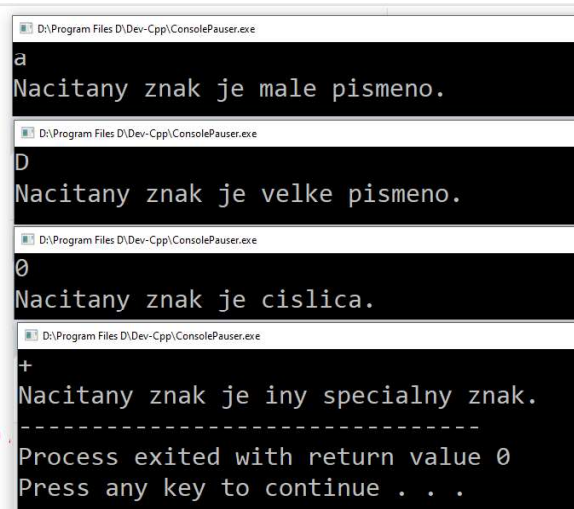
veľké písmeno (znaky od A- Z) -> 'Nacitany znak je velke pismeno.'

iný špeciálny znak -> 'Nacitany znak je iny specialny znak.'



*Správne riešenie:*

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5 char znak;
6 scanf(" %c", &znak);
7 if (znak>='a' && znak <='z') {
8 printf("Nacitany znak je male pismeno.");
9 return(0);
10 }
11 if (znak>='A' && znak <='Z') {
12 printf("Nacitany znak je velke pismeno.");
13 return(0);
14 }
15 if (znak>='0' && znak <='9') {
16 printf("Nacitany znak je cislica.");
17 return(0);
18 }
19 printf("Nacitany znak je iny specialny znak.");
20 return 0;
21 }
```



*Nápoved':*

- v tomto prípade je potrebné vhodne vytvoriť podmienku, ak je podmienka zložená z viacerých výrazov / častí spájajú sa logickými operátormi

**Časová náročnosť:** 6 min.



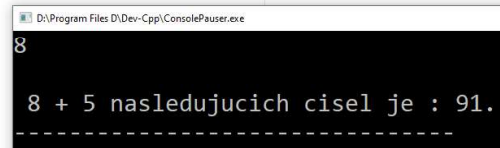
### Úloha 10: Cykly - for

V jazyku C vytvorte program, ktorý k celému číslu zadaného zo štandardného vstupu pripočíta nasledovných 5 celých číslíc. Použite cyklus for. Na obrazovku sa vypíše veta v tvare: "X + 5 nasledujucich cisel je : Y.", pričom X reprezentuje zadané číslo a Y reprezentuje výsledok.



*Správne riešenie:*

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5 int i,M, r=0;
6 scanf("%d", &M);
7 for(i=M, r=M; i<=M+5; i++)
8 r+=i;
9 printf("\n %d + 5 nasledujucich cisel je : %d.",M,r);
10 return 0;
11 }
```



*Nápoved':*

- *cyklus for obsahuje 3 časti - inicializácia riadiacej premennej, podmienka, krok; tieto 3 časti sú medzi sebou oddelené bodkočiarkou ! [vid'](#).*
- 

**Časová náročnosť:** 5 min.

### Úloha 11: Cykly - while

V jazyku C vytvorte program, ktorý bude mať za úlohu načítať reálne číslo zo štandardného vstupu tak aby patrilo do intervalu  $\langle M, N \rangle$ . Načítavanie sa opakuje pokiaľ táto podmienka nie je splnená. Hodnoty M a N budú zadané programátorom na hodnoty  $M=5.256$  a  $N=11.254$ . Na obrazovku sa vypíše veta v tvare: "Cislo z intervalu  $\langle M, N \rangle$  je: X.", pričom X reprezentuje načítané číslo z intervalu a M, N reprezentujú číselné hodnoty.

```
Select D:\Program Files D:\Dev-Cpp\ConsolePauser.exe
Zadaj cislo: 1
Zadaj cislo: 2.55487
Zadaj cislo: 12.56
Zadaj cislo: 8.256
Cislo z intervalu <5.256000,11.254000> je: 8.256000.
```

*Správne riešenie:*

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5 double M=5.256, N=11.254, cislo=0;
6 while(cislo<M || cislo>N){
7 printf("\nZadaj cislo: ");
8 scanf("%lf", &cislo);
9 }
10 printf("\nCislo z intervalu <%lf,%lf> je: %lf.",M,N,cislo);
11 return 0;
12 }
13
```

```
Select D:\Program Files D:\Dev-Cpp\ConsolePauser.exe
Zadaj cislo: 1
Zadaj cislo: 2.55487
Zadaj cislo: 12.56
Zadaj cislo: 8.256
Cislo z intervalu <5.256000,11.254000> je: 8.256000.
```

*Nápoved':*

- použite cyklus [while](#)

*Časová náročnosť:* 5 min.

## Úloha 12: Cykly - do-while

V jazyku C vytvorte program, ktorý bude mať za úlohu načítať zo štandardného vstupu len znak malého písmena. Načítavanie sa opakuje pokiaľ táto podmienka nie je splnená. Na obrazovku sa vypíše veta v tvare: "Nacitany znak je: X.", pričom X reprezentuje načítaný znak.



*Správne riešenie:*

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main() {
4 char znak;
5 do{
6 scanf(" %c",&znak);
7 }while(znak<'a' || znak>'z');
8 printf("\nNacitany znak je: %c.",znak);
9 return 0;
10 }
11
```



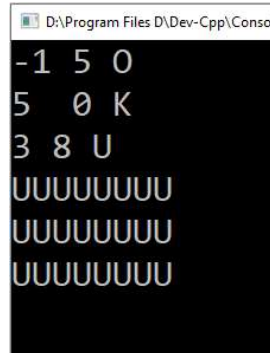
*Nápoved':*

- použite cyklus [do-while](#)

*Časová náročnosť:* 5 min.

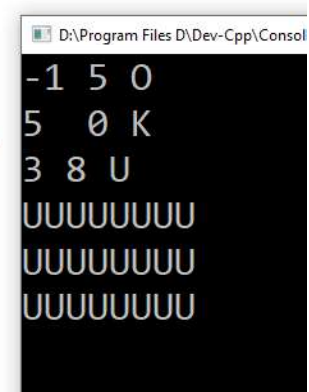
### Úloha 13: Cykly - for+for

V jazyku C vytvorte program, ktorý bude mať za úlohu vypísať na obrazovku obdĺžnik o veľkosti M riadkov a N stĺpcov zo znaku načítaného zo štandardného vstupu. Čísla M a N sa načítajú zo štandardného vstupu. Ošetrte načítanie tak aby čísla boli kladne a väčšie ako 0. Následne sa vypíše obdĺžnik na obrazovku.



*Správne riešenie:*

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main() {
4 char znak;
5 int M,N,i,j;
6 do{
7 scanf(" %d %d %c",&M, &N,&znak);
8 }while(M<=0 || N<=0);
9 for (i=0;i<M; i++) { // cyklus pre riadky
10 for (j=0;j<N; j++){ // cyklus pre stĺpce
11 printf("%c",znak);
12 }
13 printf("\n");
14 }
15 return 0;
16 }
17
```



*Nápoved':*

- použite cyklus for a v nom dalsi for, odporúčame použiť { } na ohraničenie bloku platnosti cyklu for (aj pri použití len jedného príkazu - zdrojový kód je čitateľnejší)

**Časová náročnosť:** 6 min.

### Úloha 14: Cykly - do-while + for+pole

V jazyku C vytvorte program, ktorý bude mať za úlohu vypísať na obrazovku histogram malých písmien načítavaných zo štandardného vstupu. Načítavanie znakov sa ukončí špeciálnym znakom '\*'. Následne sa vypíše histogram znakov s početnosťou väčšou ako 0.

```
Select D:\Program Files D\Dev-Cpp\ConsolePauser.exe
a s d f g a s g t r f d c *
a c d f g r s t
2 1 2 2 2 1 2 1

```

*Správne riešenie:*

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main() {
4 char znak;
5 int pole['z'-'a'+1]={0};
6 char i;
7 do{
8 scanf(" %c",&znak);
9 if (znak>='a' && znak<='z'){
10 pole[znak-'a']+=1;
11 }
12 }while(znak != '*');
13 for (i='a'; i<='z'; i++) {
14 if (pole[i-'a']>0)
15 printf(" %c",i);
16 }
17 printf("\n");
18 for (i='a'; i<='z'; i++) {
19 if (pole[i-'a']>0)
20 printf(" %d",pole[i-'a']);
21 }
22 return 0;
23 }
24
25
```

```
Select D:\Program Files D\Dev-Cpp\ConsolePauser.exe
a s d f g a s g t r f d c *
a c d f g r s t
2 1 2 2 2 1 2 1

Process exited with return value
```

*Nápoved':*

- najprv si potrebuješ pripraviť kód na načítavanie znakov zo štandardného vstupu, ak je načítaný znak '\*' načítavanie sa končí, ak sa načíta znak malého písmena tak sa uloží jeho hodnota do premennej / poľa
- výpis histogramu je podmienený počtom daného znaku, ak je väčší ako 0 vypisuje sa

*Časová náročnosť:* 10 min.

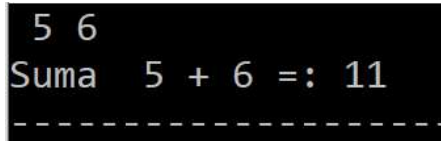
### Úloha 15: Procedúra na sčítanie čísel

V jazyku C vytvorte program, ktorý bude volať procedúru suma(). Procedúra bude bez vstupných argumentov a jej úlohou bude vypísať súčet dvoch reálnych čísel načítaných zo štandardného vstupu.

Vstup: 5 6

Výstup:

Suma 5 + 6 =11



*Správne riešenie:*

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 void suma(); // deklarácia procedúry
4
5 int main() {
6 suma(); // volanie procedúry
7 return 0;
8 }
9 void suma() { //definícia procedúry
10 int x = 0; // inicializácia premennej na zmysluplnú hodnotu
11 int y = 0; // inicializácia premennej na zmysluplnú hodnotu
12 scanf("%d %d", &x, &y);
13 int sum = x + y;
14 printf("Suma %d + %d =: %d", x, y, sum);
15 }
16
```

D:\Program Files D\Dev-Cpp\ConsolePauser.exe



*Nápoved':*

- procedúra nevracia žiadnu hodnotu, preto typ návratovej hodnoty je void
- procedúra musí byť definovaná pred volaním danej procedúry, máme dve možnosti ako to spraviť:
  - definíciu procedúry umiestnime pre funkciu main
  - definíciu procedúry umiestnime za funkciu main() ale pred funkciou main umiestnime deklaráciu procedúry void suma(); ([príklad](#))

**Časová náročnosť:** 8 min.

### Úloha 16: Funkcia pre jednoduché matematické operácie

V jazyku C vytvorte program, ktorý bude volať funkciu kalkulacka(). Funkcia bude obsahovať 3 vstupné argumenty a jej úlohou bude vykonať jednoduchú matematickú operáciu s danými číslami. 1-2 argument reprezentuje reálne čísla načítané zo štandardného vstupu a 3tí argument reprezentuje znak matematickej operácie (+, -, \*, /). Funkcia vráti výslednú hodnotu naspäť do funkcie main kde sa vypíše na obrazovku (viď ukážka). Vo funkcii main ošetríte prípad delenia nulou. V takom prípade sa vypíše veta "Nulou sa deliť neda."

Vstup: 50.5 6.352 +

Výstup: 50.5000 + 6.3520 = 56.852000

Vstup: 50.5 0 /

Výstup: Nulou sa deliť neda.

```
D:\Program Files D\Dev-Cpp\ConsolePauser.exe
21.4 25.6896 +

21.4000 + 25.6896 = 47.089600

D:\Program Files D\Dev-Cpp\ConsolePauser.exe
24.15 0 /

Nulou sa delit neda.

```

### Správne riešenie:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 double kalkulacka(double, double, char); // deklaracia funkcie
4
5 int main() {
6 double x,y=0;
7 char mat_operacia;
8 scanf("%lf %lf %c", &x, &y, &mat_operacia);
9 if (y==0 && mat_operacia=='/')
10 printf("\nNulou sa delit nedá.");
11 else
12 printf("\n %.4lf %c %.4lf = %.6lf", x,mat_operacia,y,kalkulacka(x,y,mat_operacia));
13
14 return 0;
15 }
16 double kalkulacka(double X, double Y, char MO) { //definicia funkcie
17 if (MO == '+') return (X+Y);
18 if (MO == '-') return (X-Y);
19 if (MO == '*') return (X*Y);
20 if (MO == '/') return (X/Y);
21 }
22
23
```

```
D:\Program Files D\Dev-Cpp\ConsolePauser.exe
21.4 25.6896 +

21.4000 + 25.6896 = 47.089600

D:\Program Files D\Dev-Cpp\ConsolePauser.exe
24.15 0 /

Nulou sa delit neda.

```

### Nápoved':

- funkcia môže vracat iba jednu hodnotu daného typu, preto typ návratovej hodnoty musí byť nastavený vzhľadom na výstup (funkcia môže vracat aj typ void)
- funkcia musí byť definovaná pred volaním danej funkcie, máme dve možnosti ako to spraviť:
  - definíciu funkcie umiestnime pre funkciu main
  - definíciu funkcie umiestnime za funkciu main() ale pred funkciou main umiestnime deklaráciu funkcie kalkulacka(); ([príklad](#))

Časová náročnosť: 10 min.



### Úloha 17: Funkcia na overenie číselného hesla

V jazyku C vytvorte program, ktorý bude volať funkciu kontrola\_pristupu(). Funkcia bude obsahovať 2 vstupné argumenty a jej úlohou bude vykonať načítanie a porovnanie číselného hesla. V prípade že načítané heslo nie je správne, načítanie hesla sa opakuje max N krát. Funkcia vracia hodnotu 0 alebo 1, 1 iba vtedy ak sa načítalo správne heslo. Správne heslo a počet maximálnych opakovaní vstupujú do funkcie ako vstupné argumenty. V prípade, že funkcia vráti hodnotu 1, sa na obrazovku vypíše veta "Pristup povoleny." inak "Pristup zamietnuty!".

| D:\Program Files D\Dev-Cpp\ConsolePauser.exe | D:\Program Files D\Dev-Cpp\ConsolePauser.exe |
|----------------------------------------------|----------------------------------------------|
| [1] Zadať heslo: 12345                       | [1] Zadať heslo: 125                         |
| [2] Zadať heslo: 52654                       | [2] Zadať heslo: 12345                       |
| [3] Zadať heslo: 123456                      | [3] Zadať heslo: 123654                      |
| Pristup povoleny.                            | Pristup zamietnuty!                          |
| -----                                        | -----                                        |
| Process exited with return value 0           | Process exited with return value             |
| Press any key to continue . . .              | Press any key to continue . . .              |

### Správne riešenie:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int kontrola_pristupu(int, int); // deklarácia funkcie
4
5 int main() {
6 int HESLO = 123456;
7 if (kontrola_pristupu(HESLO,3))
8 printf("\n Pristup povoleny.");
9 else
10 printf("\n Pristup zamietnuty!");
11
12 return 0;
13 }
14 int kontrola_pristupu(int HESLO, int N) { //definícia funkcie
15 int i;
16 int newHESLO=0;
17 for(i=1; i<=N; i++){
18 printf("\n [%d] Zadať heslo: ", i);
19 scanf("%d",&newHESLO);
20 if(HESLO == newHESLO)
21 return(1);
22 }
23 return(0);
24 }
25
```

| D:\Program Files D\Dev-Cpp\ConsolePauser.exe | D:\Program Files D\Dev-Cpp\ConsolePauser.exe |
|----------------------------------------------|----------------------------------------------|
| [1] Zadať heslo: 12345                       | [1] Zadať heslo: 125                         |
| [2] Zadať heslo: 52654                       | [2] Zadať heslo: 12345                       |
| [3] Zadať heslo: 123456                      | [3] Zadať heslo: 123654                      |
| Pristup povoleny.                            | Pristup zamietnuty!                          |
| -----                                        | -----                                        |
| Process exited with return value 0           | Process exited with return value             |
| Press any key to continue . . .              | Press any key to continue . . .              |

### Nápoved':

- načítanie nového hesla sa bude vykonávať v cykle, je na tebe aký cyklus si zvolíš (odporúčame for)

Časová náročnosť: 8 min.

### Úloha 18: Funkcie na súčet čísel v intervale M,N.

V jazyku C vytvorte program, ktorý bude volať funkciu `suma_cisla()`. Funkcia bude obsahovať 2 vstupné argumenty a jej úlohou bude vykonať súčet celých čísel v intervale  $\langle M, N \rangle$ . Hodnoty  $M$  a  $N$  budú vstupovať ako vstupné argumenty funkcie. Zabezpečte aby volanie funkcie prebehlo korektne, tj. musí platiť podmienka  $M \leq N$  inak sa čísla prehodia medzi sebou. Funkcia vráti súčet čísel v intervale ako návratovú hodnotu a vypíše sa veta v tvare "Súčet čísel v intervale  $\langle M, N \rangle$  je  $X$ .", kde  $M$ ,  $N$  a  $X$  budú reprezentovať čísla.

```
Select D:\Program Files D\Dev-Cpp\ConsolePauser.exe
5 6
Súčet čísel v intervale <5,6> je 11.

Process exited with return value 0
Press any key to continue . . .

D:\Program Files D\Dev-Cpp\ConsolePauser.exe
12 4
Súčet čísel v intervale <4,12> je 72.

```

*Správne riešenie:*

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int suma_cisla(int , int);
5
6 int main() {
7 int M,N;
8 scanf("%d %d", &M, &N);
9 if (M<=N) {
10 printf("Súčet čísel v intervale <%d,%d> je %d.", M,N,suma_cisla(M,N));
11 }
12 else{
13 printf("Súčet čísel v intervale <%d,%d> je %d.", N,M,suma_cisla(N,M));
14 }
15 return 0;
16 }
17
18 int suma_cisla(int M, int N) {
19 int i = M, suma=0;
20 while(i<=N) {
21 suma += i;
22 i++;
23 }
24 return(suma);
25 }
26
27
28
```

```
Select D:\Program Files D\Dev-Cpp\ConsolePauser.exe
5 6
Súčet čísel v intervale <5,6> je 11.

Process exited with return value 0
Press any key to continue . . .

D:\Program Files D\Dev-Cpp\ConsolePauser.exe
12 4
Súčet čísel v intervale <4,12> je 72.

```

*Nápoved:*

- vo funkcii `main` je potrebné ošetriť volanie funkcie tak aby interval bol správne zadany
- funkcia bude obsahovať cyklus na súčet číslíc, keďže interval je z prava aj ľava uzavretý počítajú sa aj číslíc  $M$  a  $N$
- je to možné spraviť aj pomocou rekurzcie - tj. funkcia volá samú seba ale so zmenenou hodnotou argumentu, treba dať pozor na správne ukončenie volania, aby sa funkcia nevolala do nekonečna - tiež označované ako nekonečný cyklus ([príklad](#))

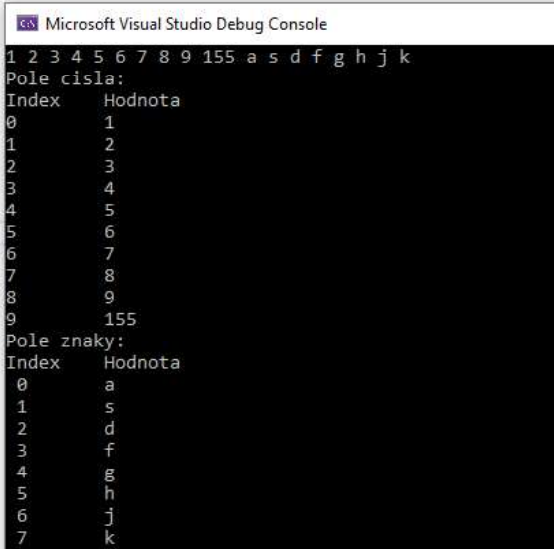
**Časová náročnosť:** 10 min.

### Úloha 19: Statické jednorozmerné pole celých čísel a pola znakov

Definujte statické jednorozmerné pole s názvom císla pre 10 celých čísel a statické jednorozmerné pole s názvom znaky pre 8 znakov. Zo štandardného vstupu (klávesnice) načítajte celé čísla do poľa číslíc a znaky do poľa znakov. Obe polia vypíšte na obrazovku spolu aj s ich indexami.

*Správne riešenie:*

```
1 #define _CRT_SECURE_NO_WARNINGS
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 /* run this program using the console pauser or add your own getch, system("pause") or input loop */
6
7 int main(int argc, char* argv[]) {
8 int císla[10];
9 int i, j;
10 char znaky[8];
11
12 for (i = 0; i < 10; i++) {
13 scanf("%d", &císla[i]);
14 }
15 //fflush(stdin);
16 for (j = 0; j < 8; j++) {
17 scanf("%c", &znaky[j]);
18 }
19 printf("Pole císla: \nIndex \t Hodnota");
20 for (i = 0; i < 10; i++) {
21 printf("\n%d \t %d", i, císla[i]);
22 }
23 printf("\nPole znaky: \nIndex \t Hodnota");
24 for (j = 0; j < 8; j++) {
25 printf("\n %d \t %c", j, znaky[j]);
26 }
27 return 0;
28 }
```



Microsoft Visual Studio Debug Console

```
1 2 3 4 5 6 7 8 9 155 a s d f g h j k
Pole císla:
Index Hodnota
0 1
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
9 155
Pole znaky:
Index Hodnota
0 a
1 s
2 d
3 f
4 g
5 h
6 j
7 k
```

*Nápoveda:*

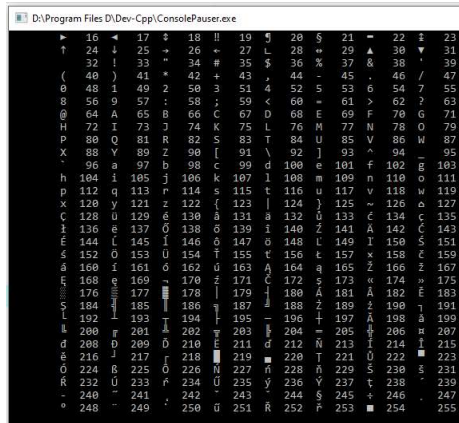
- v jazyku C sa polia indexujú od pozície 0
- pre zadefinovanie jednorozmerného statického poľa určitého typu, názvu a veľkosti použite formát

názov\_typu názov\_premennej [ veľkosť\_poľa ]; // [príklad](#)

**Časová náročnosť:** 10 minút

## Úloha 20: Statické jednorozmerné pole

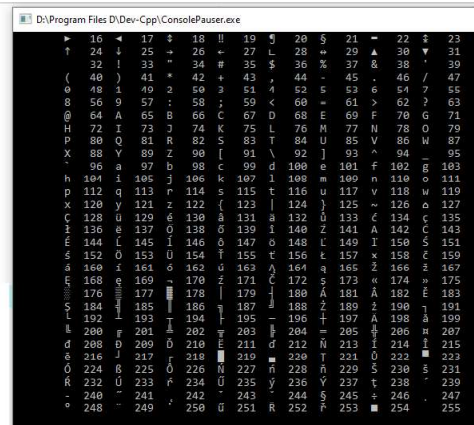
V jazyku C vytvorte program, ktorý vypíše ASCII tabuľku do 8 stĺpcov pričom každý stĺpec bude obsahovať znak a číselnú hodnotu (vypisujte znaky od hodnoty 16).



|   |     |     |     |     |     |     |     |     |
|---|-----|-----|-----|-----|-----|-----|-----|-----|
| P | 16  | 17  | 18  | 19  | 20  | 21  | 22  | 23  |
| Q | 24  | 25  | 26  | 27  | 28  | 29  | 30  | 31  |
| R | 32  | 33  | 34  | 35  | 36  | 37  | 38  | 39  |
| S | 40  | 41  | 42  | 43  | 44  | 45  | 46  | 47  |
| T | 48  | 49  | 50  | 51  | 52  | 53  | 54  | 55  |
| U | 56  | 57  | 58  | 59  | 60  | 61  | 62  | 63  |
| V | 64  | 65  | 66  | 67  | 68  | 69  | 70  | 71  |
| W | 72  | 73  | 74  | 75  | 76  | 77  | 78  | 79  |
| X | 80  | 81  | 82  | 83  | 84  | 85  | 86  | 87  |
| Y | 88  | 89  | 90  | 91  | 92  | 93  | 94  | 95  |
| Z | 96  | 97  | 98  | 99  | 100 | 101 | 102 | 103 |
| a | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 |
| b | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 |
| c | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 |
| d | 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 |
| e | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 |
| f | 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 |
| g | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |
| h | 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 |
| i | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 |
| j | 176 | 177 | 178 | 179 | 180 | 181 | 182 | 183 |
| k | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 |
| l | 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 |
| m | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 |
| n | 208 | 209 | 210 | 211 | 212 | 213 | 214 | 215 |
| o | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 |
| p | 224 | 225 | 226 | 227 | 228 | 229 | 230 | 231 |
| q | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 |
| r | 240 | 241 | 242 | 243 | 244 | 245 | 246 | 247 |
| s | 248 | 249 | 250 | 251 | 252 | 253 | 254 | 255 |

Správne riešenie:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4
5 int main() {
6 int i,j;
7 for(i=2; i<32; i++){
8 for(j=1; j<=8; j++){
9 printf("\t\t%c %4d", (char)((i*8)+j-1), ((i*8)+j-1));
10 printf("\n");
11 }
12 }
13 return 0;
14 }
```



|   |     |     |     |     |     |     |     |     |
|---|-----|-----|-----|-----|-----|-----|-----|-----|
| P | 16  | 17  | 18  | 19  | 20  | 21  | 22  | 23  |
| Q | 24  | 25  | 26  | 27  | 28  | 29  | 30  | 31  |
| R | 32  | 33  | 34  | 35  | 36  | 37  | 38  | 39  |
| S | 40  | 41  | 42  | 43  | 44  | 45  | 46  | 47  |
| T | 48  | 49  | 50  | 51  | 52  | 53  | 54  | 55  |
| U | 56  | 57  | 58  | 59  | 60  | 61  | 62  | 63  |
| V | 64  | 65  | 66  | 67  | 68  | 69  | 70  | 71  |
| W | 72  | 73  | 74  | 75  | 76  | 77  | 78  | 79  |
| X | 80  | 81  | 82  | 83  | 84  | 85  | 86  | 87  |
| Y | 88  | 89  | 90  | 91  | 92  | 93  | 94  | 95  |
| Z | 96  | 97  | 98  | 99  | 100 | 101 | 102 | 103 |
| a | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 |
| b | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 |
| c | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 |
| d | 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 |
| e | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 |
| f | 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 |
| g | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |
| h | 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 |
| i | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 |
| j | 176 | 177 | 178 | 179 | 180 | 181 | 182 | 183 |
| k | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 |
| l | 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 |
| m | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 |
| n | 208 | 209 | 210 | 211 | 212 | 213 | 214 | 215 |
| o | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 |
| p | 224 | 225 | 226 | 227 | 228 | 229 | 230 | 231 |
| q | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 |
| r | 240 | 241 | 242 | 243 | 244 | 245 | 246 | 247 |
| s | 248 | 249 | 250 | 251 | 252 | 253 | 254 | 255 |

Nápoved':

- ASCII tabuľka má 255 znakov

Časová náročnosť: 10 minút