

Napište program, ktorý bude vypisovať tabuľku mocnín čísla x od 1 do n , pričom výpočet aj výpis bude realizovať funkcia `void mocnina(double x, int n)`. Program načíta zo vstupu 2 čísla oddelené jednou medzerou a ukončené znakom konca riadku. Prvé z čísel je reálne číslo a predstavuje základ mocniny (mocnenec) a druhé číslo je celé, pričom predstavuje maximálny exponent (mocniteľ) n . Výstupom je n riadkov, pričom i -ty riadok má formát $x^i = y$, kde x je mocnenec, i je mocniteľ a y je výsledok umocnenia. Mocnenec aj výsledok vypisujte na 2 desatinné miesta. Každý riadok je ukončený znakom konca riadku.

Ukážka vstupu:

3.0 4

Výstup pre ukážkový vstup:

3.00^1 = 3.00
3.00^2 = 9.00
3.00^3 = 27.00
3.00^4 = 81.00

```
1 // uloha3-1.c -- Peter Plevko, 8.10.2019 15:03
2
3 #include <stdio.h>
4
5 void mocnina(double x, int n)
6 {
7     double s = 1, nakolku = 1;
8     int i, j;
9
10    for (i = 0; i < n; i++)
11    {
12        for (j = 1; j <= s; j++)
13        {
14            nakolku = nakolku * x;
15        }
16
17        printf("%.2lf^%.0lf = %.2lf\n", x, s, nakolku);
18        s = s + 1;
19        nakolku = 1;
20    }
21 }
22
23 int main()
24 {
25     double cislo, mocnina1;
26     scanf("%lf %lf", &cislo, &mocnina1);
27     mocnina(cislo, mocnina1);
28
29     return 0;
30 }
31 }
```

1 5.0 4

Štandardný výstup

1

Napište program, ktorý bude vypisovať tabuľku násobkov čísla x od 1 do n , pričom výpočet aj výpis bude realizovať funkcia `void nasobok(double x, int n)`. Výpočet sa bude robiť len pomocou sčítovania, bez použitia násobenia. Program načíta zo vstupu 2 čísla oddelené jednou medzerou a ukončené znakom konca riadku. Prvé z čísel je reálne číslo a predstavuje prvý činiteľ a druhé číslo je celé, pričom predstavuje maximálny druhý činiteľ n . Výstupom je n riadkov, pričom i -ty riadok má formát $x * i = y$, kde x je prvý činiteľ, i je druhý činiteľ a y je ich súčin. Prvý činiteľ aj súčin sa má vypisovať na 2 desatinné miesta. Každý riadok je ukončený znakom konca riadku.

Ukážka vstupu:

3.0 2

Výstup pre ukážkový vstup:

3.00 * 1 = 3.00
3.00 * 2 = 6.00

```
1 #define _CRT_SECURE_NO_WARNINGS
2 #include <stdio.h>
3 #include <math.h>
4 int main()
5 {
6     double vysledok, cislo, kamil;
7     int i;
8     scanf("%lf %lf",&cislo,&kamil);
9     for (i = 1; i <= kamil; i++)
10     {
11         vysledok = cislo * i;
12         printf("%.2lf * %d = %.2lf \n", cislo, i, vysledok);
13     }
14 }
15
16 return 0;
17
18 }
```

1 3.0 2

Štandardný výstup

1

Kompilácia

Napište program, ktorý zo štandardného vstupu načíta dve celé čísla a vypíše všetky prvočísla z daného intervalu. V programe použite funkciu `zisti_prvocislo(int cislo)`, ktorej parametrom je zisťované číslo a návratovou hodnotou je číslo 1 ak zisťované číslo je prvočíslo, inak -1.

Program načíta zo vstupu 2 čísla oddelené jednou medzerou a ukončené znakom konca riadku (hranice intervalu). Výstupom programu budú prvočísla z daného intervalu, každé v samostatnom riadku. Ošetríte hranice intervalu. V prípade, ak v zadanom intervale neexistuje prvočíslo program vypíše správu `Prvocislo neexistuje`. Každý riadok je ukončený znakom konca riadku.

Ukážka vstupu:

1 4

Výstup pre ukážkový vstup:

2
3

```
1 // uloha3-3.c -- Peter Plevko, 9.10.2019 08:49
2
3 #include <stdio.h>
4
5 int main(void)
6 {
7
8     int a, b, i, j, flag;
9
10
11     scanf("%d", &a);
12
13
14     scanf("%d", &b);
15
16
17
18     if (a <= 0) a = 1;
19
20     for (i = a; i <= b; i++) {
21         if (a<0) {break;}
22
23         if (i == 1 || i == 0)
24             continue;
25
26
27         flag = 1;
28
29         for (j = 2; j <= i / 2; ++j) {
30             if (i % j == 0) {
31                 flag = 0;
32                 break;
33             }
34         }
35
36
37         if (flag == 1)
```

1 1 50

1

Napište program, ktorý v prvom riadku načíta celé číslo n predstavujúce počet vstupov. Potom zo vstupu prečíta n riadkov, každý obsahujúci dvojicu reálnych čísel: hodinovú mzdu a počet odpracovaných hodín za týždeň. Dvojica reálnych čísel je na vstupe oddelená vždy jednou medzerou a ukončená znakom konca riadku. Pre každú dvojicu čísel program zavolá funkciu `double tyzdenna_mzda(double h_mzda, double hod)`, ktorá vráti mzdu za týždeň. Pre každú z n dvojíc vypíše program jeden riadok ukončený znakom konca riadku a obsahujúci správu `Hod. mzda: m Euro/hod, hodin: h, spolu: s Euro`, kde m je hodinová mzda, h je počet odpracovaných hodín a s je celková mzda za týždeň.

Výpočet mzdy za týždeň sa vykoná takto: do 40 odpracovaných hodín sa vypláti základná hodinová mzda, za hodiny medzi 40 a 60 sa vypláti jedenaplnásobok hodinovej mzdy a za hodiny nad 60 sa vypláti dvojnásobok hodinovej mzdy. Na konci program vypíše správu `Celkova mzda: c Euro`, kde c je celková suma, ktorá sa má vyplátiť pre všetkých n dvojíc. Správa je ukončená znakom konca riadku. Všetky čísla na výstupe sú vypísané na 2 desatinné miesta. Použite konštanty, čím sa vyhniete používaniu "magických" čísel.

Ukážka vstupu:

```
3
20 38
22 48
24 68
```

Výstup pre ukážkový vstup:

```
Hod. mzda: 20.00 Euro/hod, hodin: 38.00, spolu: 760.00 Euro
Hod. mzda: 22.00 Euro/hod, hodin: 48.00, spolu: 1144.00 Euro
Hod. mzda: 24.00 Euro/hod, hodin: 68.00, spolu: 2064.00 Euro
Celkova mzda: 3968.00 Euro
```

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <stdlib.h>
4  #include <string.h>
5  #include <stdio.h>
6  #include <ctype.h>
7  #include <stdio.h>
8
9  #include <stdio.h>
10
11 double tyzdenna_mzda(double h_mzda, double hod)
12 {
13     double dokopy=0;
14     if (hod < 40) dokopy = h_mzda * hod;
15     if (hod > 40 && hod < 60) dokopy = (hod - 40) * (h_mzda * 1.5) + h_mzda * 40;
16     if (hod > 60) dokopy = (hod - 60) * (h_mzda * 2) + (h_mzda * 1.5) * 20 + h_mzda * 40;
17     printf("Hod. mzda: %.2lf Euro/hod, hodin: %.2lf, spolu: %.2lf Euro\n", h_mzda, hod, dokopy);
18     return dokopy;
19 }
20
21 int main()
22 {
23     int pocetvstupov;
24     double mzda, hod, spolu=0;
25     scanf("%d", &pocetvstupov);
26     for (int i = 0; i < pocetvstupov; i++)
27     {
28         scanf("%lf %lf", &mzda, &hod);
29         spolu = spolu + tyzdenna_mzda(mzda, hod);
30     }
31     printf("Celkova mzda: %.2lf Euro", spolu);
32     return 0;
33 }
34 }
```

```
1  5
2  20 38
3  22 48
4  24 68
5  3 120
6  4 130
```

Napište program, v ktorom načítate celé číslo n a postupne n celých čísel. V programe použite funkciu `int armstrong(int x)`. Táto funkcia zistí či zadané čísla sú Armstrongove. Číslo je Armstrongove ak suma číslíc daného čísla umocnená na najvyšší rad tohto čísla je rovná zadnému číslu. Ak zisťované číslo je Armstrongove funkcia vráti 1, inak 0.

Príklady Armstrongových čísel:

$7 = 7^1$
 $371 = 3^3 + 7^3 + 1^3$ (27 + 343 + 1)
 $8208 = 8^4 + 2^4 + 0^4 + 8^4$ (4096 + 16 + 0 + 4096)

Ukážka vstupu:

3
 371
 123
 8208

Výstup pre ukázkový vstup:

371 je Armstrongove cislo
 123 nie je Armstrongove cislo
 8208 je Armstrongove cislo

```

1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <math.h>
4
5  int armstrong(int x)
6  {
7      int zvysook = 0, vysledok = 0, mocnina = 0, cislo1, j = 0, cislo, i, pocetcisel;
8      cislo = x;
9      cislo1 = cislo;
10     while (cislo1 > 0)
11     {
12         mocnina = mocnina + 1;
13         cislo1 = cislo1 / 10;
14     }
15     cislo1 = cislo;
16     for (j = 0; j < mocnina; j++)
17     {
18         zvysook = cislo1 % 10;
19         vysledok = vysledok + pow(zvysook, mocnina);
20         cislo1 = cislo1 / 10;
21     }
22     if (vysledok == cislo) return 1;
23     else return 0;
24
25
26
27
28
29 }
30
31
32 int main()
33 {
34     int zvysook=0,vysledok=0,mocnina=0, cislo1, j = 0, cislo, i, pocetcisel;
35     scanf("%d", &pocetcisel);
36     for (i = 0; i < pocetcisel; i++)
37     {

```

```

1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <math.h>
4
5  int armstrong(int x)
6  {
7      int zvysook = 0, vysledok = 0, mocnina = 0, cislo1, j = 0, cislo, i, pocetcisel;
8      cislo = x;
9      cislo1 = cislo;
10     while (cislo1 > 0)
11     {
12         mocnina = mocnina + 1;
13         cislo1 = cislo1 / 10;
14     }
15     cislo1 = cislo;
16     for (j = 0; j < mocnina; j++)
17     {
18         zvysook = cislo1 % 10;
19         vysledok = vysledok + pow(zvysook, mocnina);
20         cislo1 = cislo1 / 10;
21     }
22     if (vysledok == cislo) return 1;
23     else return 0;
24
25
26
27
28
29 }
30
31
32 int main()
33 {
34     int zvysook=0,vysledok=0,mocnina=0, cislo1, j = 0, cislo, i, pocetcisel;
35     scanf("%d", &pocetcisel);
36     for (i = 0; i < pocetcisel; i++)
37     {

```

1 450

2 50

3 51

4 52

5 53

6 54

7 55

8 56

9 57

10 58

11 59

12 60

13 61

14 62

15 63

16 64

17 65

18 66

19 67

20 68

21 69

22 70

23 71

Štandardný výstup

1

Napište program, na zisťovanie reverzného čísla. Program načíta zo vstupu číslo x ukončené znakom konca riadku. V programe použijete funkciu `long reverzne_cislo(long x)` ktorá vráti reverzné číslo k číslu x . Výstupom programu bude vrátené reverzné číslo.

Program rozšírite tak, že bude načítat všetky čísla zo vstupu. Počet čísel na vstupe nie je známy pred spustením programu. Využite návratovú hodnotu funkcie `scanf`. Ku každému načítanému číslu vytvorí reverzné číslo a navyše zistí, či načítané číslo je palindróm a vypíše správu: číslo x je palindrom, alebo číslo x nie je palindrom, kde x je zisťované číslo. Správa je nasledovaná znakom konca riadku.

Ukážka vstupu:

```
12345
12321
```

Výstup pre ukázkový vstup:

```
54321
Cislo 12345 nie je palindrom
12321
Cislo 12321 je palindrom
```

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <math.h>
4
5  long reverzne_cislo(long x)
6  {
7      int n, r = 0, t;
8      n = x;
9      t = n;
10
11     while (t != 0)
12     {
13         r = r * 10;
14         r = r + t % 10;
15         t = t / 10;
16     }
17
18     return r;
19 }
20 int main()
21 {
22     int n, r = 0, t;
23
24
25     while (scanf("%d", &n) >= 1)
26     {
27
28
29         if (reverzne_cislo(n) == n)
30         {
31             printf("%d\n", reverzne_cislo(n));
32             printf("Cislo %d je palindrom\n", n);
33         }
34         else {
35             printf("%d\n", reverzne_cislo(n));
36             printf("Cislo %d nie je palindrom\n", n);
37         }
```

Kompilácia

```
1  0
2  1
3  2
4  3
5  4
6  5
7  6
8  7
9  8
10 9
11 10
12 11
13 12
14 13
15 14
16 15
17 16
18 17
19 18
20 19
21 20
22 21
```

Štandardný výstup

```
1
```

Napište program, v ktorom vo funkcii `main` budete mať deklarované dve premenné `c1`, `c2` typu `char` a premennú `p_c` typu ukazovateľ na `char`. Zo štandardného vstupu postupne načíta znaky do premennej `c1`, pokiaľ nenačíta znak `'*'`. Každý načítaný znak prekopíruje do premennej `c2` pričom nebude použitý príkaz `c2 = c1`. Nepoužívajte ani žiadne iné premenné ako `c1`, `c2` a `p_c`. Koľkými spôsobmi sa to dá urobiť? Výstupom programu je jeden riadok pre každý načítaný znak. Každý z týchto riadkov je ukončený znakom konca riadku a má mať takýto formát:

`c1: x (y), c2: z (p), p_c: q (r)`

kde `x` je obsah premennej `c1` a `y` adresa tejto premennej. Podobne, `z` je obsah premennej `c2` a `p` jej adresa. `q` predstavuje hodnotu premennej, na ktorú ukazuje ukazovateľ `p_c` a `r` je hodnota `p_c`.

Ukážka vstupu:

jazyk C*

Výstup pre ukázkový vstup:

```
c1: j (0x7ffffac467b3e), c2: j (0x7ffffac467b3f), p_c: j (0x7ffffac4
c1: a (0x7ffffac467b3e), c2: a (0x7ffffac467b3f), p_c: a (0x7ffffac4
c1: z (0x7ffffac467b3e), c2: z (0x7ffffac467b3f), p_c: z (0x7ffffac4
c1: y (0x7ffffac467b3e), c2: y (0x7ffffac467b3f), p_c: y (0x7ffffac4
c1: k (0x7ffffac467b3e), c2: k (0x7ffffac467b3f), p_c: k (0x7ffffac4
c1:  (0x7ffffac467b3e), c2:  (0x7ffffac467b3f), p_c:  (0x7ffffac4
c1: C (0x7ffffac467b3e), c2: C (0x7ffffac467b3f), p_c: C (0x7ffffac4
```

```
1 // uloha3-7.c -- Peter Plevko, 4.11.2019 21:10
2
3 #include <stdio.h>
4
5 int main()
6 {
7     char c1,c2,*p_c;
8
9     while((c1=getchar())!='*')
10    {
11        p_c=&c2;
12        *p_c=c1;
13        //dalsie riesenie
14        //p_c=&c1;
15        //c2=*p_c;
16        printf("c1: %c (%p), c2: %c (%p), p_c: %c (%p)\n",c1,&c1,c2,&c2,*p_c,p_c);
17    }
18    return 0;
19 }
```