

Úloha 3.1: Napíšte funkciu `int parne(int x[], int pocetx, int y[])`, ktorá skopíruje všetky párne čísla z poľa `x` do poľa `y` v poradí v akom sa nachádzajú v poli `x` a vráti počet prvkov poľa `y`. Argument `pocetx` určuje počet prvkov poľa `x`. Môžete predpokladať, že argument `y`, bude mať dostatočnú veľkosť pre všetky párne prvky.

Napr.:

`x = {4, 7, 1, 3, 2, 5, 6}`

`pocetx = 7`

`pocety = parne(x, pocetx, y); // volanie funkcie`

`pocety: 3 // vypis vysledku`

`y: {4, 2, 6}`

```
#include <stdio.h>
int parne(int x[], int pocetx, int y[])
{
    // sem napis svoje riesenie
}
int main()
{
    int x[7] = {4, 7, 1, 3, 2, 5, 6};
    int pocetx = 7;
    int y[7];
    int i, pocety = parne(x, pocetx, y);
    printf("pocety = %d\n", pocety);
    for (i = 0; i < pocety; i++)
    {
        if (i > 0)
            printf(", ");
        printf("%d", y[i]);
    }
    printf("\n");
    return 0;
}
```

Úloha 3.2: Napíšte funkciu `int nasobky(int x[], int pocetx, int y[], int k)`, ktorá z poľa `x` prekopíruje do poľa `y` všetky násobky čísla `k` (`k >= 0`), v poradí v akom sa nachádzajú v poli `x` a vráti počet prvkov poľa `y`. Argument `pocetx` určuje počet prvkov poľa `x`. Môžete predpokladať, že argument `y`, bude mať dostatočnú veľkosť pre všetky násobky čísla `k`, ktoré sú v poli `x`.

Napr.:

`x = {4, 7, 10, 1, 3, 9, 2, 5, 8, 6}`

`pocetx = 10`

`pocety = nasobky(x, pocetx, y, 2); // volanie funkcie`

`pocety: 5 // vypis vysledku`

`y: {4, 10, 2, 8, 6}`

Úloha 3.3: Napíšte funkciu `int delitele(int x[], int pocetx, int y[], int k)`, ktorá z poľa `x` prekopíruje do poľa `y` všetky delitele čísla `k`, v poradí v akom sa nachádzajú v poli `x` a vráti počet prvkov poľa `y`. Argument `pocetx` určuje počet prvkov poľa `x`. Môžete predpokladať, že argument `y`, bude mať dostatočnú veľkosť pre všetky delitele čísla `k`, ktoré sú v poli `x`.

Napr.:

`x = {4, 7, 10, 2, 3, 9, 6, 5, 8, 12}`

`pocetx = 10`

`pocety = delitele(x, pocetx, y, 24); // volanie funkcie`

`pocety: 6 // vypis vysledku`

`y: {4, 2, 3, 6, 8, 12}`

Úloha 3.4: Napíšte program, ktorý načíta z klávesnice tri čísla oddelené medzerami a vypíše ich súčet. Na uloženie prvých dvoch čísel použite smerníky na typ `integer` (pred načítaním si pre uloženie čísel dynamicky alokujte pamäť), tretie číslo bude uložené do obvyčajnej premennej typu `integer`. Výstupom programu bude jeden riadok vo formáte: `cislo1 + cislo2 + cislo3 = sucet`

Ukážka vstupu:

`1 2 3`

Výstup pre ukážkový vstup:

`1 + 2 + 3 = 6`

Úloha 3.5: Napíšte program, ktorý zo vstupu načíta celé číslo `n` a alokuje v pamäti blok `n` položiek pre znaky. Potom zo štandardného vstupu načíta `n` znakov a vypíše ich odzadu.

Ukážka vstupu:

`4`

`ahojky`

Výstup pre ukážkový vstup:

`joha`

Úloha 3.6: Napíšte program, ktorý zistí počet jednotlivých písmen v každom riadku súboru. Vstupom programu je jeden riadok obsahujúci meno súboru. Výstupom je histogram výskytu písmen zapísaný v prehľadnej tabuľke, kde prvý riadok bude obsahovať všetky písmená abecedy prehľadne oddelené. Každý ďalší riadok bude obsahovať číslo riadku a vždy pod písmenami budú zarovnané počty výskytov tohto písmena v jednotlivých riadkoch súboru (nerozlišujte medzi veľkými a malými písmenami). Počty výskytov uveďte ako najviac dvojciferné celé číslo predchádzané jednou medzerou. Všetky riadky výstupu budú ukončené znakom konca riadku.

Ukážka vstupu:

subor.txt

Ukážka obsahu súboru subor.txt:

aDefDghI Hlk mPrstT V.

ABBA bc DEED DFD hih MopMp RsR WvxWW YxzY.

Výstup pre ukážkový vstup:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	R	S	T	U	V	W	X	Y	Z
1	0	0	2	1	1	1	2	2	1	1	0	1	0	0	1	1	1	2	0	1	0	0	0	0	
2	3	1	4	2	1	3	2	1	0	0	0	2	0	1	2	2	1	0	0	1	3	1	2	1	