

# Задачи

## Увод в програмирането

12 ноември 2018 г.

### Правила

- Правилното декомпозиране на функции е много важно. Вашите функции трябва да са кратки (по-малко от 20 реда!) И всяка трябва да изпълнява една единствена, ясна задача.
- Имената на променливи трябва да са описателни – да обясняват за какво служи дадената променлива. Примерно за име на човек подходящо име е `personName`, а неподходящо име е `a1`.

### Задачи

1. !!!!!!! Ако не сте решили всички задачи от предишните пъти решете първо тях !!!!!!! :
  - <https://github.com/tozka/fmi-kn-8/blob/master/up-ex3/tasks-22-10-2018.pdf>
  - <https://github.com/tozka/fmi-kn-8/blob/master/up-ex4/tasks-29-10-2018.pdf>
  - <https://github.com/tozka/fmi-kn-8/blob/master/up-ex5/tasks-05-11-2018.pdf>
2. Напишете функция, наречена `nameDiamond`, която приема низ като параметър и го отпечатва във формат "диамант", както е показано по-долу. На пример да отпечата името `CHRIS`

```
C
CH
CHR
CHRI
CHRIS
HRIS
RIS
IS
S
```

3. Намерете грешките, оправети ги и обяснете (с коментари) как ги оправяте.

a)

```
int sum (int x, int y) {
    int result;
    result = x + y;
}
```

b)

```
int sum (int n) {
    if (0 == n)
        return 0; else
        n = n + n;
}
```

**c)** in main program:

```
double x = 1E10;  
cout << "square of 1E10 = " << square (x) << endl;
```

```
int square (int x)    {  
    return x * x;  
}
```

**d)**

```
int main() {  
    int i = 0;  
    for(i = 0; i < 3; i++); {  
        std::cout << i << std::endl;  
        continue;  
    }  
    return 0;  
}
```

**e)**

```
#include<iostream>
```

```
void Main() {  
    int a = 10;  
    std::cout << a  
}
```

**f)**

// Упражнението е да се направи функция, която взема вход и ако входът е четен, се извършва определено изчисление. Ако е нечетен, се извършва друго. Тогава новата стойност се използва и продължава, докато стигне 1.

```
#include <iostream>  
using namespace std;  
int count_something(int x) {  
    int counter;  
  
    for(x; x == 1; ) {  
        if(x % 2 == 0) {  
            x = x / 2;  
            counter++;  
        }  
        else if(x % 2 != 0) {  
            x = x * 3 + 1;  
            counter++;  
        }  
    }  
    return counter;  
}  
  
int main() {
```

```

int x=10, counter;
count_something(x);
cout << counter << endl;
return 0;
}

```

**g)**

```

int isPrime(int n){
    int i = 0;
    for(i=0; i < n; i++)
        if(n%i == 0)
            return 0;
    return 1;
}

void check (int numbers[], int finish) {
    for(i=0; i<finish; i++)
        if(!isPrime(i)) {
            std::cout << numbers[i] << "is prime\r\n";
        }
}

```

**h)**

```

#include <iostream>

int main(int argc, const char *argv[]) {
    char passWordInput[100];
    void mainMenu();
    cin.getline(passWordInput, 200);
    cout << strlen(passWordInput);
    return 0;
}

void mainMenu() {
    cout << "Your password must be at least 8 characters long and" << endl;
    cout << "contain at least one character from the following categories." << endl;
}

```

4. Напишете функция, която при подаден символен низ:  
 (като се броят само буквите от азбуката)  
 (изхода не е нужно да са лексикографски сортирани)  
 (проверката за буквите да не е case-sensitive т.е A = a)  
 може да използвате помощни масиви
- а. връща колко пъти се среща всяка буква
- Вход: "abcaadc"
- Изход: a -> 2  
 b -> 1  
 c -> 2  
 d -> 1

б. принтира последователност от уникални символи от масива

т.е. всички символи изкарани по веднъж

Вход: "abcadc"

Изход: "abcd"

5. Да се напише функция, която връща колко монети от 1 лв., 20 ст., 5 ст. и 1 ст. са нужни за изплащане на дадена сума до 100 лв., зададена като дробно число лева, с до две цифри след десетичната запетая. Да се избират възможно най-едри монети.

Пример: с 2,77 лв

Вход: 2.77

Изход: 2 x 1 lv + 3 x 20 st + 3 x 5 st + 2 x 1 st

6. Да се напише функция която проверява дали подаден символен низ е валиден имейл.

Преди @ са позволени знаци [a-z],[A-Z],[0-9], тире, подчертавка

След @ са позволени [a-z],[A-Z] и една единствена точка

Пример: example\_exalple-12345@abc.com евалиден имейл