

**W ramach uzupełnienia zadań z zajęć można wykonać zadania oznaczone na żółto.**

### **Zadanie 13 (zadanie wykonane w czasie zajęć)**

W programie będącym prostym kalkulatorem (napisanym na pierwszych zajęciach) i działającym w taki sposób, że obliczenia można wykonywać do momentu podania przez użytkownika 0, które oznacza koniec pracy, należy zastąpić minimum 5 funkcjami. Pierwsza funkcja o deklaracji void menu() ma wyświetlać menu programu, pozostałe funkcje mają realizować operacje matematyczne: float dodawanie(float, float), float odejmowanie(float, float), float mnożenie(float, float), float dzielenie(float, float). Funkcje należy zdefiniować poza funkcją main.

### **Zadanie 14 (zadanie wykonane w czasie zajęć)**

Napisz funkcję o definicji void zamiana1(int, int), która dokona zamiany dwóch przekazanych przez wartość liczb, np.: int a = 4; int b = 8; na a = 8 i b = 4. Funkcję należy wywołać w funkcji main w taki sposób:

```
cout << „podaj liczbę 1” << endl; cin >> a;
```

```
cout << „podaj liczbę 2” << endl; cin >> b;
```

```
zamiana1(a, b);
```

```
cout << „podaj liczbę 1” << endl; cin >> a;
```

```
cout << „podaj liczbę 2” << endl; cin >> b;
```

Dla porównania, stwórz funkcję void zamiana2(int &, int &), która przekaże do zamiany dwie liczby zapisane do zmiennych a i b przez referencję. Funkcję zamiana2 należy wywołać w takiej samej funkcji main jak powyżej. Porównaj wyniki. Co zauważyłeś?

### **Zadanie 15 (zadanie wykonane w czasie zajęć)**

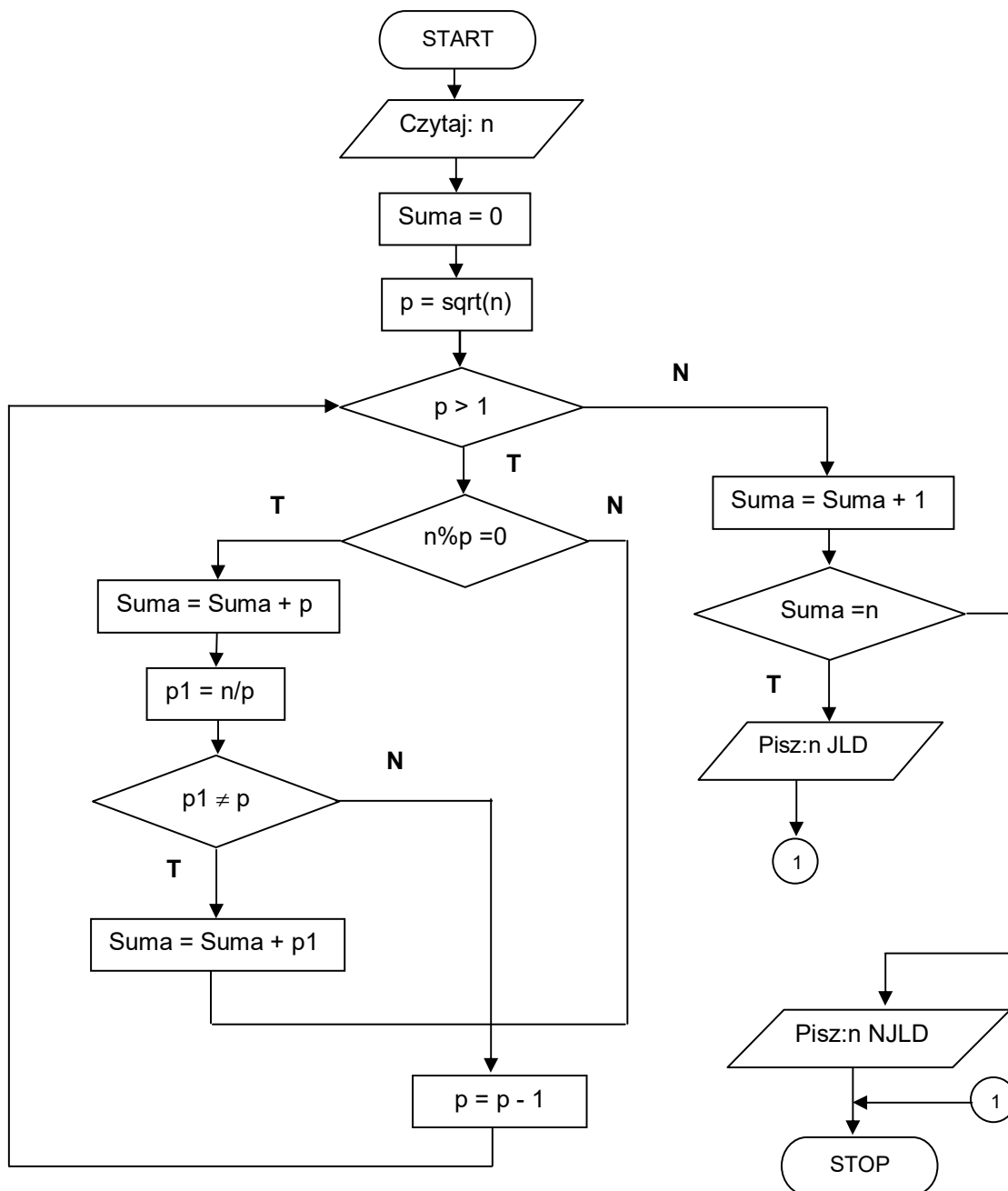
Napisz program na zbadanie, czy wczytana z klawiatury dodatnia liczba całkowita  $n$  jest liczbą doskonałą.

UWAGA 1: Liczba doskonała to taka, dla której suma jej dzielników (nie licząc samej liczby  $n$ ), jest równa tej liczbie. Przykładami liczb doskonałych są liczby 6 i 28.

UWAGA 2: Wystarczy sprawdzić podzielniki nie większe niż  $\sqrt{n}$ ;

Program należy napisać w taki sposób, aby umożliwić wielokrotne jego wykonanie bez konieczności powrotu do edytora.

Możesz wykorzystać poniższy algorytm lub zaproponować własny:



### **Zadanie 16 (zadanie do samodzielnego wykonania)**

Napisz program wyszukujący liczby pierwsze z przedziału  $[2...n]$ . Liczbę naturalną  $n$  wczytaj z klawiatury.

Program należy napisać w taki sposób, aby umożliwić wielokrotne jego wykonanie bez konieczności powrotu do edytora.

### **Zadanie 17 (zadanie do samodzielnego wykonania)**

Napisz program obliczający, ile różnych podzbiorów k-elementowych można utworzyć ze zbioru n elementów, czyli liczymy:

$$m = \frac{n!}{k!(n-k)!}$$

Program należy napisać w dwóch wersjach:

- bez użycia funkcji
- z użyciem dwóch funkcji: na obliczenie silni oraz na obliczenie wyniku.

### **Zadanie 18 (zadanie do samodzielnego wykonania)**

Napisz program na obliczenie pierwiastka funkcji  $f(x)$  w przedziale  $\langle a, b \rangle$  metodą bisekcji, gdzie:

$$f(x) = e^{-1,5x} - 0,3x^2,$$

$$a = 0;$$

$$b = 1;$$

Pierwiastek należy obliczyć z dokładnością  $\varepsilon$ .

Dokładność  $\varepsilon$  należy wczytać z klawiatury. Oblicz i wyświetl również liczbę iteracji.

Program należy napisać w dwóch wersjach:

- bez użycia funkcji
- z użyciem dwóch funkcji: na obliczenie wartości funkcji  $f(x)$  oraz poszukującej pierwiastek metodą bisekcji

Dane testowe:  $\varepsilon = 1e-7$ .

Wynik: pierwiastek = 0,917481; liczba iteracji: 24

### **Zadanie 19 (zadanie wykonane w czasie zajęć)**

Napisz program obliczający silnię dla podanej liczby n w sposób iteracyjny i rekurencyjny. Wykaż poprawność działania funkcji.

Dla podanej liczby n = 3 poprawny wynik to:  $1*2*3 = 6$

W tym celu użyj prototypów funkcji:

```
int sil_iter(int n); // wersja iteracyjna
```

```
int sil_rek(int n); // wersja rekurencyjna
```

### **Zadanie 20 (zadanie wykonane w czasie zajęć)**

Napisz program obliczający sumę dla podanej liczby  $n$  w sposób iteracyjny i rekurencyjny. Wykaż poprawność działania funkcji.

Przykładowo, dla podanej liczby  $n=3$ , poprawny wynik to:  $1+2+3 = 6$

W tym celu użyj prototypów funkcji:

```
int sil_iter(int n); // wersja iteracyjna
```

```
int sil_rek(int n); // wersja rekurencyjna
```