

Sprawozdanie z projektu

Adam Zygmunt P02

Projekt 1

Spis treści

Wstęp	2
Opis problemu	2
Schemat blokowy algorytmu	3
Algorytm w pseudokodzie	4
Algorytm z komentarzami	5

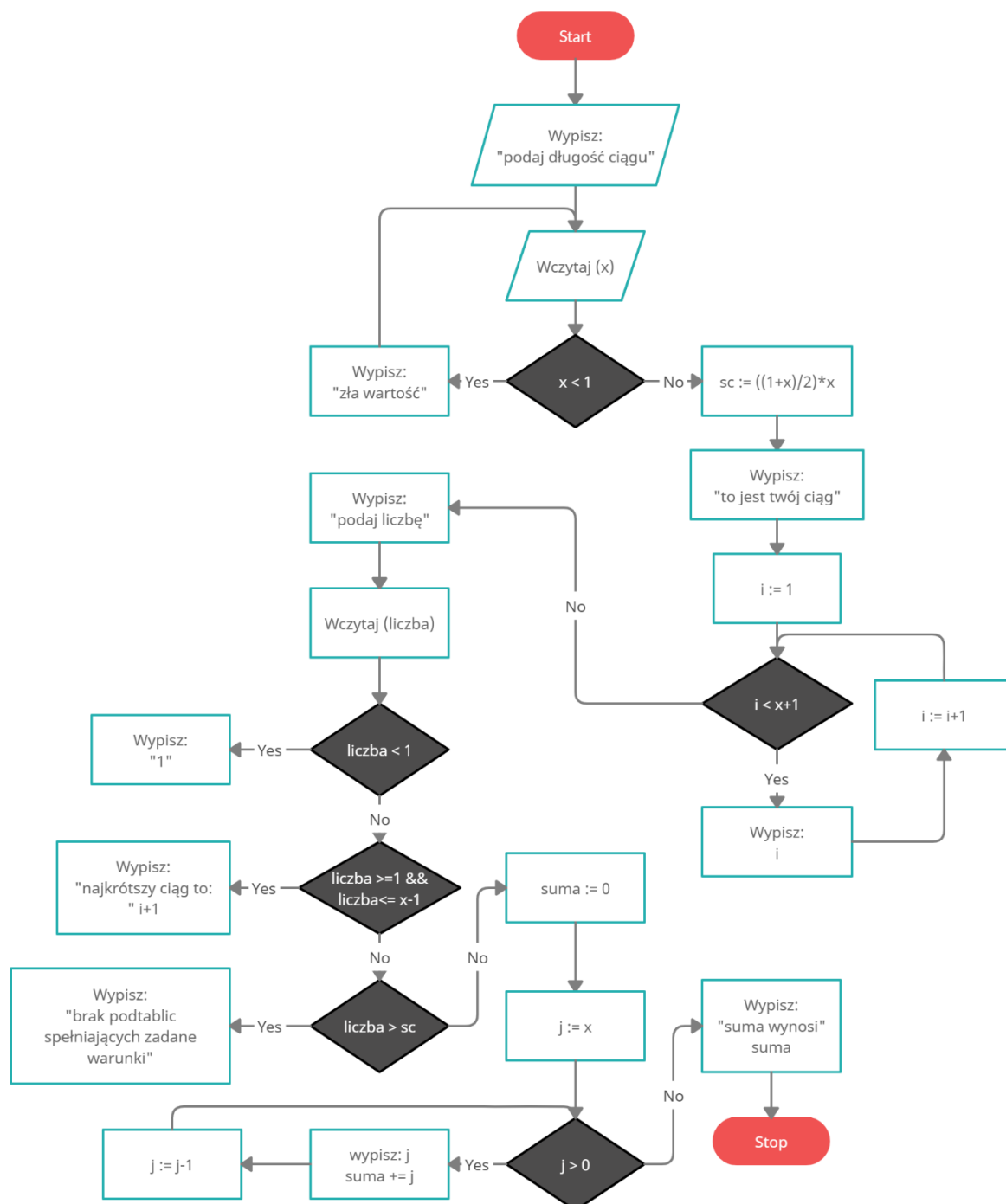
Wstęp

Dokument ten jest sprawozdaniem z pierwszego w tym semestrze projektu z przedmiotu Algorytmy i struktury danych. Napisany przeze mnie program powstał w języku C++, w kompilatorze CodeBlocks.

Opis problemu

Należy utworzyć ciąg liczb naturalnych. Powinna zostać wprowadzona zmienna „x”, która będzie zarówno liczbą elementów tego ciągu, jak i ostatnim jego wyrazem. Następnie należy wprowadzić dowolną liczbę, oznaczoną pod zmienną „liczba”. Celem programu jest zwracanie najkrótszego ciągu, którego suma wyrazów jest większa od zadanej liczby. Zwracany ciąg jest częścią ciągu zadanego na początku.

Schemat blokowy algorytmu



Algorytm w pseudokodzie

Wypisz „podaj długość ciągu”;

Wczytaj (x);

Dopóki $x < 0$:

Wypisz " podaj nieujemne x "

$sc = ((1+a)/2)*a$;

Wypisz ("to jest twój ciąg\n");

Dla $i = 1$ do x , powtarzaj:

wypisz i ;

Wypisz "podaj liczbę "

Wczytaj (liczba)

Jeśli $liczba < 0$:

wypisz „1”;

Jeśli nie, czy: $liczba < x$:

wypisz $x+1$;

Jeśli nie, czy: $liczba > sc$:

wypisz "brak podtablic spełniających warunki";

Jeśli nie:

$suma = 0$;

jeśli $suma < liczba$:

$j = x$;

dopóki $j > 0$ wykonuj:

wypisz j ;

$suma = suma + j$;

$j = j-1$

jeśli $suma > liczba$:

przerwij;

wypisz: $suma$;

Algorytm z komentarzami

/*Zadanie 14.

Dla zadanego ciagu liczb naturalnych, znajdz dlugosc najkrotszego ciagu,
ktorego suma jest wiesz niz zadana liczba.

Przyklad.

Wejscie: 1, 2, 3, 4, 5, 6, 7, 8, k = 20

Wyjscie: 6, 7, 8

Wejscie: 1, 2, 3, 4, 5, 6, 7, 8, k = 7

Wyjscie: 8

Wejscie: 1, 2, 3, 4, 5, 6, 7, 8, k = 21

Wyjscie: 5, 6, 7, 8

Wejscie: 1, 2, 3, 4, 5, 6, 7, 8, k = 40

Wyjscie: Brak podtablic spelniajacych zadane warunki*/

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    /* zadeklarowanie zmiennych: liczba (w tresci zadania k, ale nazwalem "liczba", aby uniknac  
    pomylek),
```

```
    x-liczba elementow ciagu, suma - na biezaco bedzie sumowac wyrazy ciagu od konca, sc-suma  
    wszystkich wyrazów ciagu*/
```

```
    int  liczba, x, suma, sc ;
```

```
    // wyswietlenie komunikatu proszacego o podanie dlugosci ciagu - x
```

```
    cout << "podaj jak dlugi chcesz ciag (a1=1, r=1): ";
```

```

// wprowadzenie wartosci x
cin >> x;

// petla uniemozliwiajaca wprowadzenie dlugosci ciagu < 1
while( x<1 )
{
    cout << "podano zla wartosc, sprobuj ponownie ";
    cin >> x;
}
// wzór na sume wyrazów ciagu
sc = ((1+x)/2)*x;

// utworzenie tablicy dynamicznej
int *tab= new int[x];

// wyswietlenie zadanego ciagu
cout<<"to jest twoj ciag"<< endl;

for(int i=1; i < x+1 ; i++)
{
    cout<< i << " ";
}
// komunikat proszacy o podanie liczby k
cout << "\n" << "podaj liczbe : ";

// wprowadzenie liczby
cin >> liczba;

// jezeli dana liczba jest mniejsza od 1, to najkrotszym ciagiem bedzie jednoelementowy ciag
a(liczba)=1

if(liczba<0) cout << "1" << endl;

```

```

// jezeli liczba miesci sie w zakresie <0, x-1>, wyswietlenie jednoelementowego ciagu a(n)=liczba+1
else if(liczba>=0 && liczba<=x-1)
{
    cout << "\n" << "najkrotszy ciag to :" << liczba+1 << endl; ;
}

// jezeli liczba jest wieksza od sumy wyrazów zadanego ciagu, wyswietlenie komunikatu o braku
rozwiązania

else if ( liczba > sc) cout << "Brak podtablic spełniających zadane warunki ";

// w przeciwnym wypadku
else
{
    // wprowadzenie pomocniczej zmiennej j
    int j;

    //ustawiam początkowa sume jako 0
    suma = 0;

    // dopoki liczba jest wieksza od wartosci "suma", dodawanie do niej kolejnych elementow
    zbioru (od konca)

    // oraz wyswietlanie ich w postaci ciagu wyjsciowego

    for(j=x; j>0; j--)
    {
        cout << j << " ";
        suma += j;
        if(suma>liczba)break;
    }

    // kontrolne wyswietlenie sumy ciagu wyjsciowego
    cout << "\n suma wynosi :" << suma;

}

return 0;
}

```