

## Sprawozdanie z informatyki nr. 1

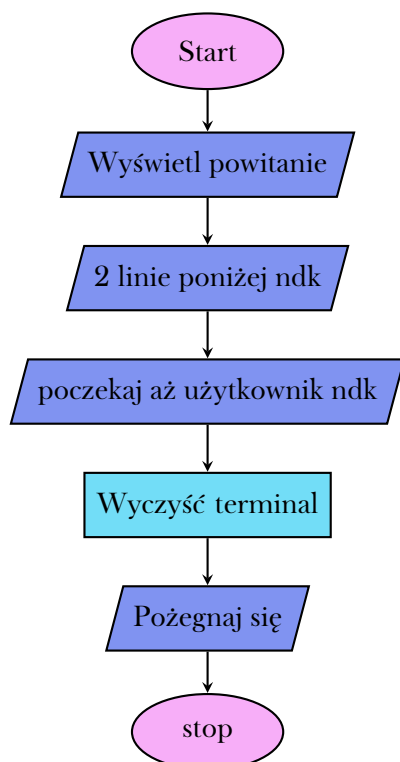
## Treść zadanie

## Programownie część pierwsza

- 1.0 - Wyświetl napis powitalny, 2 wiersze poniżej ndk, cls, pożegnanie
- 1.1 - Wartość wyrażenia, 2 miejsca po przecinku:  $\frac{2*3+17}{9}$  (wynik 2, 56)
- 1.2 - Wartość wyrażenia, 3 miejsca po przecinku:  $\frac{4^2+2*4*7+7^2}{5+37\div 4}$  (wynik 8, 491)
- 1.3 - Powitanie, cls, pole pow. i objętość sześcianu o boku 5,875 cm (2mpp) (p=207,09; obj=202,78)
- 1.4 - Pole objętość i suma długości krawędzi sześcianu o boku 7.225 cm (ze stałą)  
 $P = 313, 20$ ;  $O = 377, 15$ ;  $S = 86, 70$
- 1.5 - Średnia arytmetyczna z 3 liczb - stałe (7,12,16) oraz napis NDK Średnia=11,67
- 1.6 - Pole, objętość i suma długości krawędzi prostopadłościanu o bokach podanych przez użytkownika (2mpp)
- 1.7 - Pole, objętość i suma długości krawędzi walca o promieniu i wysokości podanych przez użytkownika (pi jako stała)
- 1.8 - Średnia arytmetyczna z 3 liczb podanych przez użytkownika z jego imieniem.
- 1.9 - Pole prostokąta, ze sprawdzeniem danych (czy boki są większe od 0)
- 1.10 - Czy dwie liczby podane przez użytkownika są podzielne przez siebie – pierwsza przez drugą – uwaga na warunek podzielności
- 1.11 - Równanie liniowe w pełnej postaci ( $ax + b = c$ )
- 1.12 - Równanie  $ax^2 + bx + c = d$  kwadratowe lub liniowe – wyniki na dole ekranu
- 1.13 - Wyświetlanie maksymalnej liczby z trzech podanych liczb całkowitych ze sprawdzeniem poprawności danych
- 1.14 - Sprawdzenie możliwości skonstruowania trójkąta z trzech odcinków, których długości są losowymi liczbami rzeczywistymi, losowanymi z przedziału od <1,10> z jednym miejscem po przecinku (wyświetla liczby na górze i komunikat na dole ekranu)

## Proponowane rozwiązania

## 1.0



D: brak

S: brak

```

1  cout << "Witam w programie" << endl<< endl;
2  cout << "Nacisnij dowolny kalwiesz";
3  getch();
4  system("cls");
5  cout << "do widzenia";
6  getch();
  
```

Figure 1: 1.0 flowchart

## 1.1

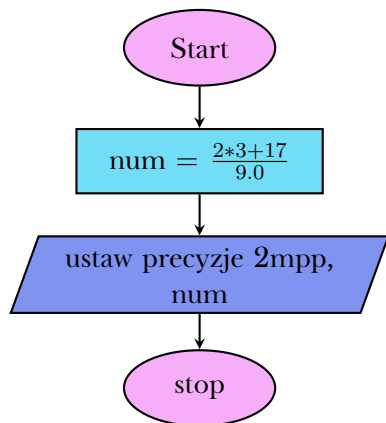


Figure 2: 1.1 flowchart

D:?

S: num Wartość wyrażenia  $\in 2,56$

```

1  double num = (2*3+17)/9.0;
2  cout << setprecision(3) << num;
  
```

## 1.2

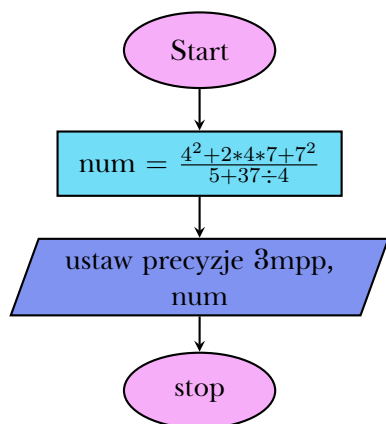


Figure 3: 1.2 flowchart

D:?

S: num wartość wyrażenia  $\in 8,491$

```

1  float num = (pow(4,2)+2*4*7+pow(7,2))/(5+37/4.0);
2  cout << setprecision(4) << num;
  
```

## 1.3

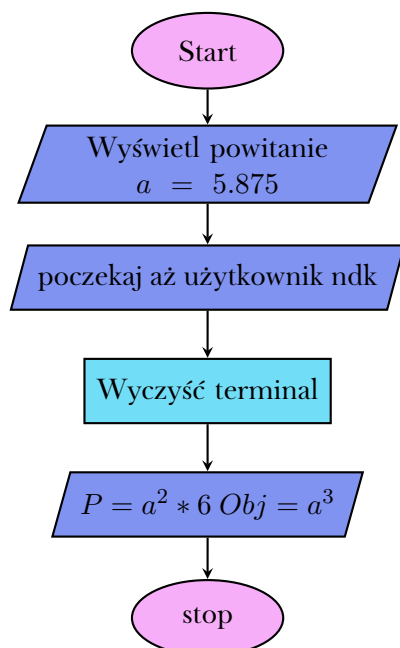


Figure 4: 1.3 flowchart

D:a bok  $\in 5.875$

S: P, Obj wymiary  $\in \mathbb{R}$

```

1  string pow ("witam w programie");
2  char* pow_arr = new char[pow.length()];
3  strcpy(pow_arr, pow.c_str());
4  for(n=0;n<=pow.length()n++){
5      cout << pow_arr[n-1];
6      Sleep(250);}
7  Sleep(2000);
8  system("cls");
9  float bok = 5.875;
10 float pole = 6.0*bok*bok;
11 float obj = 1.0*bok*bok*bok;
12 cout<<setprecision(3)<<"p="<<pole<<"obj="<< obj;
  
```

## 1.4

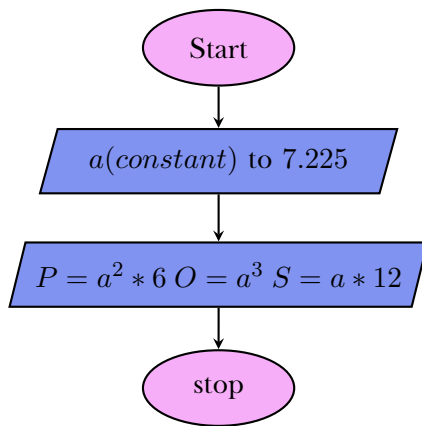


Figure 5: 1.4 flowchart

D:  $a \text{ bok} \in \{7.225\}$

S:  $P, O, S \text{ wymiary} \in \{313, 20; 377, 15; 86, 70\}$

```

1 cout <<fixed<< setprecision(2)<<\
2 "dla boku:"<<a << endl <<\
3 "P="<< a*a*6<< endl <<\
4 "O=" << a*a*a << endl<<\
5 "S="<< a*12;
  
```

## 1.5

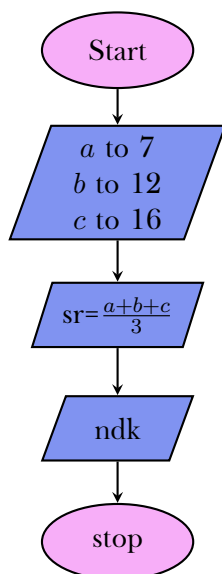


Figure 6: 1.5 flowchart

D:  $a, b, c \text{ parametry} \in \{7; 12; 16\}$

S:  $sr \text{ średnia parametrów} \in \{11, 67\}$

```

1 double sr = (a+b+c)/3.0;
2 cout << fixed <<setprecision(2)<<\
3 "Srednia: " << sr << endl
  
```

## 1.6

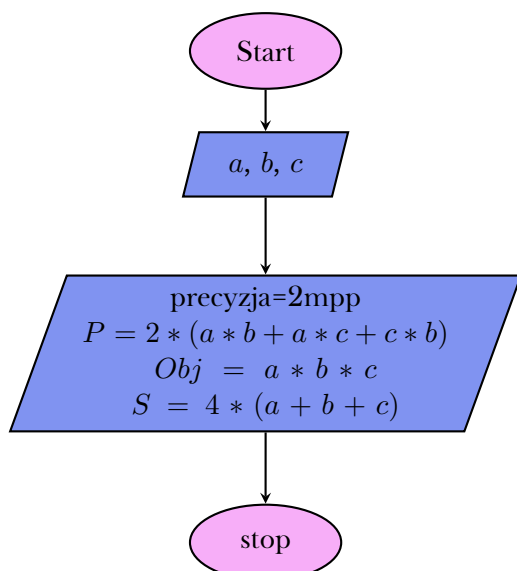


Figure 7: 1.6 flowchart

D:  $a, b, c \text{ parametry} \in \mathbb{R} > 0$

S:  $P, Obj, S \text{ wymiary} \in \mathbb{R}$

```

1 float a;
2 float b;
3 float c;
4 cout<<"Proszę podać bok a:"; cin >> a;
5 cout <<"Proszę podać bok b:"; cin >> b;
6 cout <<"Proszę podać bok c:"; cin >> c;
7 cout<<fixed<<setprecision(2)<<\
8 "Objetosc jest równa="<<a*b*c<<endl<<\
9 "pole jest równe="<<2*(a*b+a*c+b*c)<<endl<<\
10 "Suma dlugosci krawedzi to="<<4*(a+b+c);
  
```

## 1.7

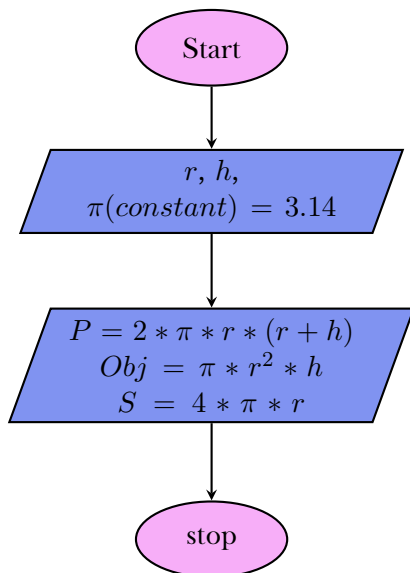


Figure 8: 1.7 flowchart

D: r, h parametry  $\in \mathbb{R} > 0$

S: P, Obj, S wymiary  $\in \mathbb{R}$

```

1 float r;
2 float h;
3 const float pi=3.14;
4 cout<<"Proszę podać promień:"; cin >> r;
5 cout <<"Proszę podać wysokość:"; cin >> h;
6 cout<<fixed<<setprecision(2)<<\
7 "Objętość jest równa="<<pi*r*r*h<<endl<<\
8 "pole jest równe="<<2*pi*r*(r+h)<<endl<<\
9 "Suma długości krawędzi to="<<4*pi*r;
  
```

## 1.8

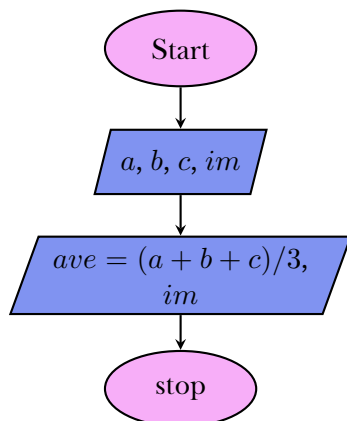


Figure 9: 1.8 flowchart

D: a, b, c, im; Liczy i imię użytkownika  $\in \mathbb{R}$

S: ave średnia  $\in \mathbb{R}$

```

1 float a;
2 float b;
3 float c;
4 string name;
5 cout<<"Podaj imię użytkownika:"; cin >> name;
6 cout<<"Proszę podać pierwszą liczbę:"; cin >> a;
7 cout<<"Proszę podać drugą liczbę:"; cin >> b;
8 cout<<"Proszę podać trzecią liczbę:"; cin >> c;
9 float ave = (a+b+c)/3;
10 cout<<"Średnia jest równa ="<<ave<<endl;
11 cout<<"Imię użytkownika to: "<<name<<endl;
  
```

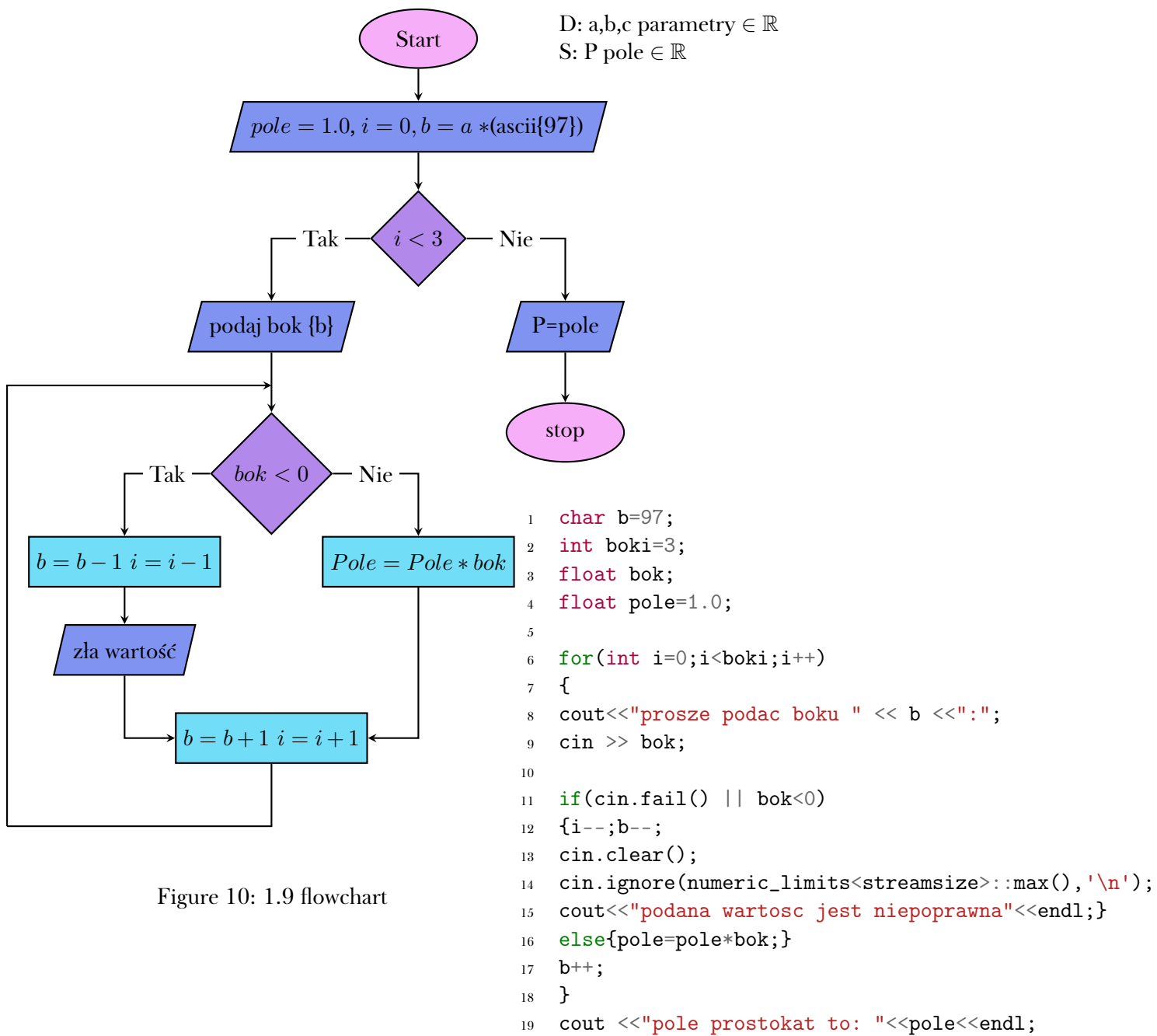
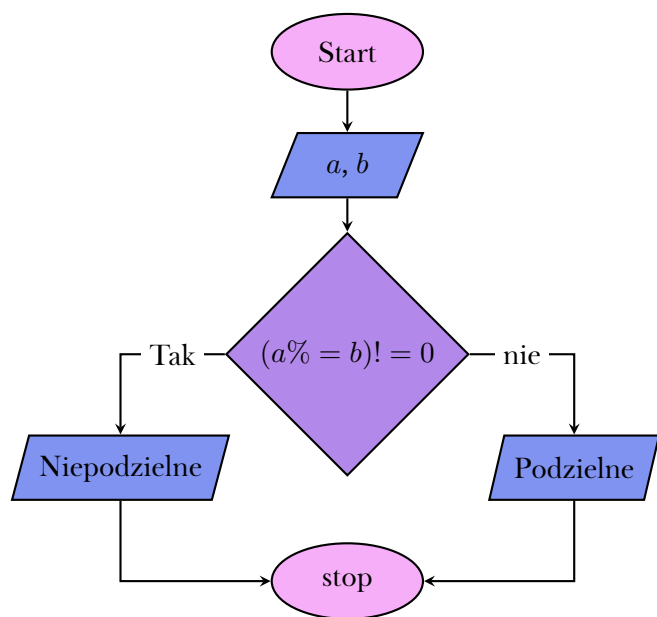


Figure 10: 1.9 flowchart

## 1.10



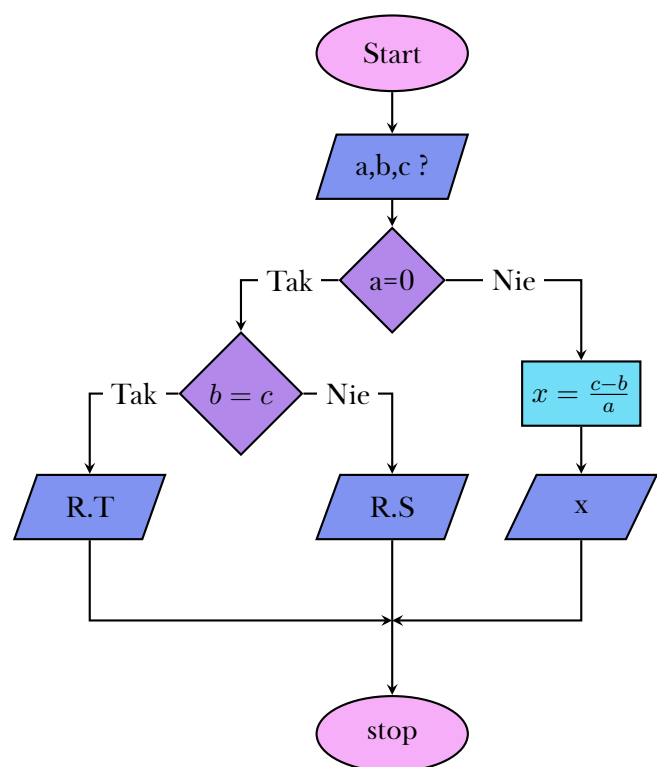
D:  $a, b$  liczby  $\in \mathbb{R}$  czy są w rzeczywistości  
 S:  $x$  podzielność  $\in$  prawda/fałsz

```

1  int num1;
2  int num2;
3  cout<<"prosze podac pierwszą liczbę:"; cin >> num1;
4  cout<<"prosze podac drugą liczbę:"; cin >> num2;
5  num1 %= num2;
6  if(num1!=0){cout<<"liczby są niepodzielne"<< endl;
7  }else{cout<<"liczby są podzielne"<< endl;}
  
```

Figure 11: 1.10 flowchart

## 1.11



Równanie:  $ax + b = c$

D:  $a, b, c$  wyznaczniki  $\in \mathbb{R}$

S:  $x$  rozwiązanie równania  $\in \mathbb{R}$

```

1  float a;
2  float b;
3  float c;
4  cout<<"prosze podać a:"; cin>>a;
5  cout<<"prosze podać b:"; cin>>b;
6  cout<<"prosze podać c:"; cin>>c;
7  if(a==0){
8      if(b==c){cout<<"Równanie tożsamościowe"<<endl;
9      }else{cout<<"równanie sprzeczne"<< endl;}}
10 else{a=(c-b)/a;cout<<"x jest równy: "<< a<<endl;}
  
```

Figure 12: 1.11 flowchart

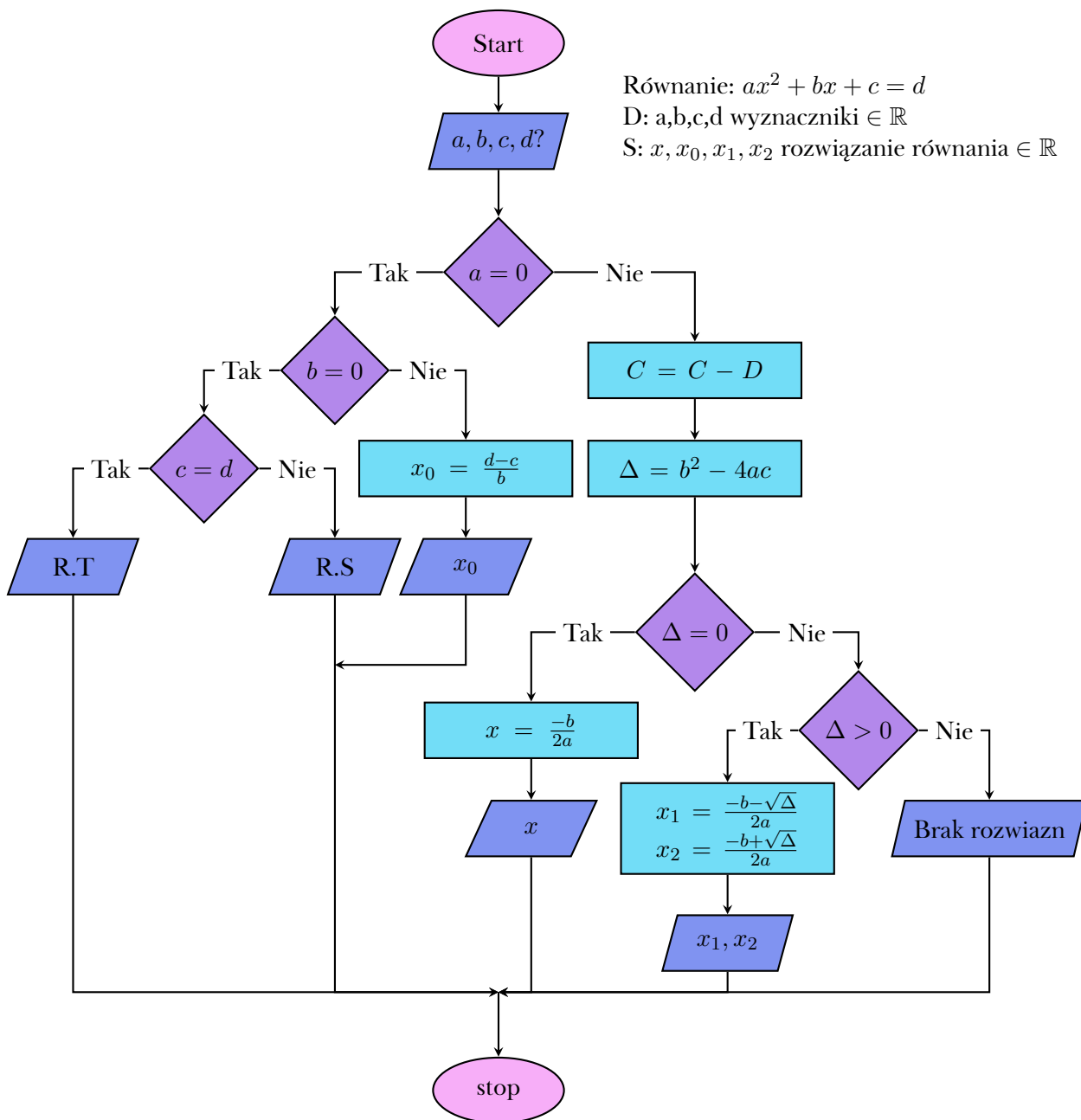
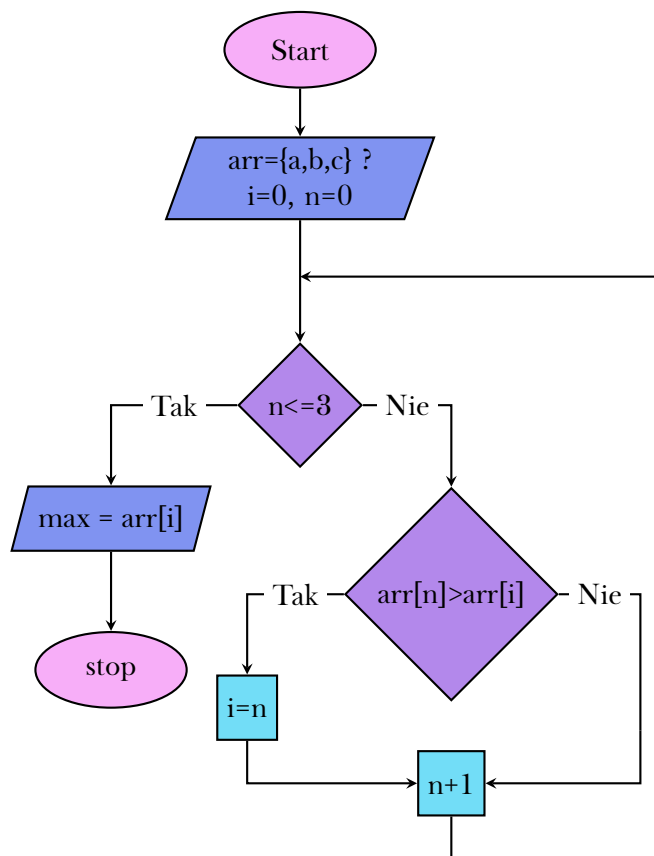


Figure 13: 1.12 flowchart

```

1  float a; cout<<"prosze podać a:"; cin>>a; float b; cout<<"prosze podać b:"; cin>>b;
2  float c; cout<<"prosze podać c:"; cin>>c; float d; cout<<"prosze podać d:"; cin>>d;
3  float delta; gotoxy(0,25);
4  if(a==0){
5      if(b==0){
6          if(c==d){cout<<"Równanie tożsamośćowe"<<endl;}
7          else{cout<<"równanie sprzeczne"<< endl;}}
8      else{b=(d-c)/b;cout<<"x jest równy: "<<b<<endl;}
9  }else{ c=c-d;
10     float delta=pow(b,2)-(4.0*a*c);
11     if(delta==0){a=(-b)/(2*a));cout<<"x jest równy: "<< a <<endl;
12     }else{
13         if(delta>0){cout<<"x1 jest równy: "<< ((-b-sqrt(delta))/(2*a))<<endl;
14             cout<<"x2 jest równy: "<<((-b+sqrt(delta))/(2*a))<<endl;
15         }else{cout<<"brak rozwiązań"<<endl;}}
  
```

## 1.13



D: a,b,c liczby  $\in \mathbb{N}_0$   
 S: max największa liczba  $\in \mathbb{N}_0$

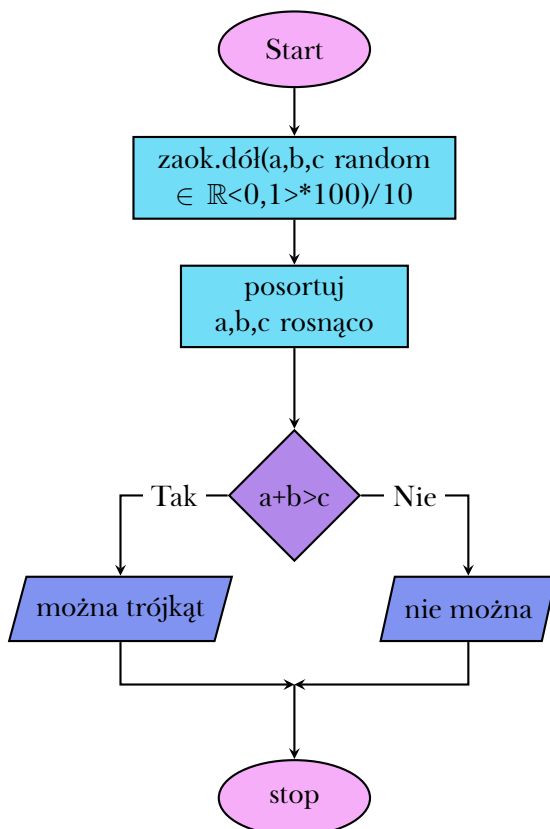
```

1  int n = 3;
2  int t;
3  int arr[n];
4  int i=0;
5  while(i<n)
6  {   cout << "prosze podać " << i+1 << " liczbe:";
7      cin >> t;
8      arr[i]={t};
9      i++;}
10 n = 0;
11 i = 0;
12 while(n<sizeof(arr)/sizeof(arr[0]))
13 {   if(arr[n]>arr[i]){i=n;}
14     n++;}
15 cout << "max number is: " << arr[i];

```

Figure 14: 1.13 flowchart

## 1.14



D: brak  
 S: możliwość zbudowania trójkąta {tak, nie}

```

1  int n = 3;
2  float a[n];
3  float b;
4
5  for(int i=0;i<n;i++)
6  {a[i-1]={ceil((rand()/ double(RAND_MAX))*100)/10+1};}
7
8  for(int i=0;i<n;i++)
9  {for(int j=0;j<n;j++)
10 {if(a[i-1]<a[j-1])
11 {b=a[i-1];a[i-1]=a[j-1];a[j-1]=b;}}}
12
13 gotoxy(0,0);
14 print_arr(a,n);
15 gotoxy(0,28);
16 if(a[-1]+a[0]>a[1]){cout<<"tak"<<endl;}
17 else{cout<<"nie"<<endl;}

```

Figure 15: 1.14 flowchart



## python

## autohotkey

### 1.0

```
1 print("Witaj w programie\n\n\n")
2 print("Naciśnij dowolny klawisz")
3 m.getch()
4 os.system("cls")
```

```
5 print("Do widzenia")
6 m.getch()
```

```
1 %-----Auto hot ket-----
```

### 1.1

```
1 print(round((2*3+17)/9,2))
2 m.getch()
```

```
1 MsgBox Round((2*3+17)/9,2),"równanie"
```

### 1.2

```
1 print(round((4**2+2*4*7+7**2)/(5+37/4),3))
```

```
1 MsgBox Round((4*4+2*4*7+7*7)/(5+37/4.0),3),"równanie"
```

### 1.3

```
1 print("Witaj w programie\n\n\n")
2 print("Naciśnij dowolny klawisz")
3 m.getch()
4 os.system("cls")
5
```

```
6 bok=5.875
7 print(\
8 "pole pow:", round(bok**2*6,2) ,\
9 "obj:", round(bok**3,2))
```

```
1 %-----Auto hot ket-----
```

### 1.4

```
1 bok=7.225
2 print(\
3 "pole pow:", round(bok**2*6,2) ,\
4 "\nobj:", round(bok**3,2),\
5 "\nsuma krawedzi:",round(bok*12,2))
```

```
1 bok := 7.225
2 p:=Round(bok*bok*6,2)
3 o:=Round(bok*bok*bok,2)
4 s:=Round(bok*12,2)
5 MsgBox ( "P=" p "`nO=" o "`nS=" s ),"równanie"
```

### 1.5

```
1 a=7
2 b=12
3 c=16
4
```

```
5 print("naciśnij dowolny klawisz")
6 m.getch()
7 print("Średnia:",round((7+12+16)/3,2))
```

```
1 %-----Auto hot ket-----
```

### 1.6

```
1 a = float(input("Prosze podac bok a:"))
2 b = float(input("Prosze podac bok b:"))
3 c = float(input("Prosze podac bok c:"))
4 print(\
```

```
5 "obj:", round(a*b*c,2),\
6 "\npole pow:", round(2*(a*b+a*c+b*c),2) ,\
7 "\nsuma krawedzi:",round(4*(a+b+c),2))
```

```
1 %-----Auto hot ket-----
```

## 1.7

```
1 r = float(input("Proszę podać bok r:"))
2 h = float(input("Proszę podać bok h:"))
3 pi = float(3.14)
4 print(\
5 "obj:", round(pi*r*r*h,2),\
6 "\npole pow:", round(2*pi*r*(r+h),2) ,\
7 "\nsuma krawedzi:",round(4*pi*r,2))
1 %-----Auto hot ket-----
```

## 1.8

```
1 name = input("Proszę podać imię użytkownika:")
2 a = float(input("Proszę podać liczbę a:"))
3 b = float(input("Proszę podać liczbę b:"))
4 c = float(input("Proszę podać liczbę c:"))
5 print("Średnia:", (a+b+c)/3,"\nnimię użytkownika:",name)
1 %-----Auto hot ket-----
```

## 1.9

```
1 bok=1.0
2 pole=1.0
3 i=0
4 while i < 3:
5     try:
6         bok = float(input("Proszę podać bok:"))
7     except ValueError:
8         print("podana wartosc jest niepoprawna\n")
9         continue
10    if bok < 0:
11
12        print("podana wartosc jest niepoprawna\n")
13    else:
14        pole=pole*bok
15        i += 1
16    print("pole prostokat to:",pole)
1 %-----Auto hot ket-----
```

## 1.10

```
1 num1 = int(input("Proszę pierwszą liczbę: "))
2 num2 = int(input("Proszę pierwszą liczbę: "))
3 num1 %= num2
4 if num1 == 0:print("podzielne")
5 else:print("nie podzielne")
1 %-----Auto hot ket-----
```

## 1.11

```
1 a = float(input("Proszę podać a: "))
2 b = float(input("Proszę podać b: "))
3 c = float(input("Proszę podać c: "))
4
5 if a == 0:
6     if b==c:print("Równanie tożsamościowe")
7     else:print("Równanie sprzeczne")
8 else:print("x jest równy:", (c-b)/a)
1 %-----Auto hot ket-----
```

## 1.12

```
1 a = float(input("Proszę podać a: "))
2 b = float(input("Proszę podać b: "))
3 c = float(input("Proszę podać c: "))
4 d = float(input("Proszę podać d: "))
5 if a == 0:
6     if b==0:
7         if c==d:print("Równanie tożsamościowe")
8         else:print("Równanie sprzeczne")
9     else:print("x jest równy:", (d-c)/b)
10 else:
11     c=c-d
12     delta = b**2-(4*a*c)
13     print("delta:", delta)
14     if delta==0: print("x jest równy:", (-b)/(2*a))
15     else:
16         if delta>0:
17             print("x1 jest równy:", (-b-mm.sqrt(delta))/(2*a))
18             print("x2 jest równy:", (-b+mm.sqrt(delta))/(2*a))
19         else: print("brak rozwiązań")
1 %-----Auto hot ket-----
```

## 1.13

```
1 temp={}
2 i=0
3 while i < 3:
4     try:
5         num1 = int(input("Proszę podać liczbę: "))
6     except ValueError:
7         print("podana wartość jest niepoprawna")
8         continue
9     temp[i] = num1
10     i=i+1
11     i=0
12 while i<3:
13     if temp[0]<temp[i]:temp[0]=temp[i]
15     print("Maksymalna liczba to:",temp[0])
1 %-----Auto hot ket-----
```

## 1.14

```
1 temp={}
2 i=0
3 while i < 3:
4     temp[i]=secrets.randbelow(90)/10+1
5     i=i+1
6 print(temp)
7 i=0
8 while i<3:
9     if temp[i]>=sum(temp.values())-temp[i]:i=4
10    i=i+1
11 if i>=4:print("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\nnie można")
12 else:print("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\nmożna")
1 %-----Auto hot ket-----
```

*Programy użyte do wykonania zadań*

---

LaTeX, google chrome, overleaf, dev c++, python, visual studio code, notepad++, git, github, Sumatra PDF, autohotkey, Total commander

*Wnioski i uwagi*

---

Zadanie mi się bardzo podobało i nie mam żadnych uwag.