

Sprawozdanie z informatyki nr 2

Treść zadanie

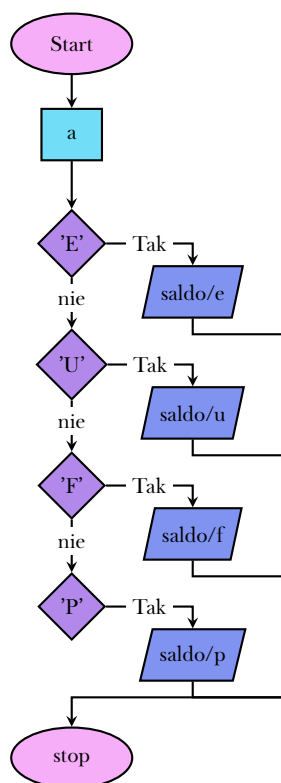
Programownie część Druga

- 2.1 - Program przeliczający waluty
- 2.2 - Program obliczający pola i obwody 4 figur geometrycznych foremnych
- 2.3 - Program sprawdzający ile jest cyfr w liczbie całkowitej z przedziału <0,miliard>
- 2.4 - Liczby podzielne z przedziału
- 2.5 - Średnia arytmetyczna z n liczb podanych przez użytkownika
- 2.6 - Wyświetlenie i zliczenie wszystkich naturalnych liczb trzycyfrowych w których suma cyfr wynosi n
- 2.7 - Przerób program 2.6 tak, aby działał w pętli
- 2.8 - Średnia arytmetyczna z n liczb dwucyfrowych, dodatnich podanych przez użytkownika
- 2.9 - Program obliczający NWD i NWW 2 liczb a i b
- 2.10 - Tabliczka mnożenia od-do
- 2.11 - Szukanie minimalnej liczby z ciągu liczb dwucyfrowych podawanych przez użytkownika
- 2.12 - Program obliczający NWD z 2 liczb(odejmowanie)
- 2.13 - Obliczanie n! dla liczby naturalnej $n \geq 0$ – algorytm iteracyjny
- 2.14 - Obliczanie wartości n-tego wyrazu ciągu Fibonacciego ($n \geq 1$) – algorytm iteracyjny
- 2.15 - Sprawdzanie czy liczba n ($n > 0$) jest liczbą pierwszą
- 2.16 - Suma cyfr liczby całkowitej n ($n > 0$)

Proponowane rozwiązania

2.1

dis()



D:
Saldo, kwota pieniędzy $\in \mathbb{R}$
c, waluta $\in \{u, f, p, e\}$
W: strumień, saldo w innych walutach $\in \mathbb{R}$

```

1  dis(char a)
2  {switch (a)
3  {case 'E':
4      cout<<"EUR:"<<saldo/e<<endl;
5      break;
6  case 'U':
7      cout<<"USD:"<<saldo/u<<endl;
8      break;
9  case 'F':
10     cout<<"CHF:"<<saldo/f<<endl;
11     break;
12 case 'P':
13     cout<<"PLN:"<<saldo/p<<endl;
14     break;
15 }}
16

```

Figure 1.0: funcion dis() flowchart

Main

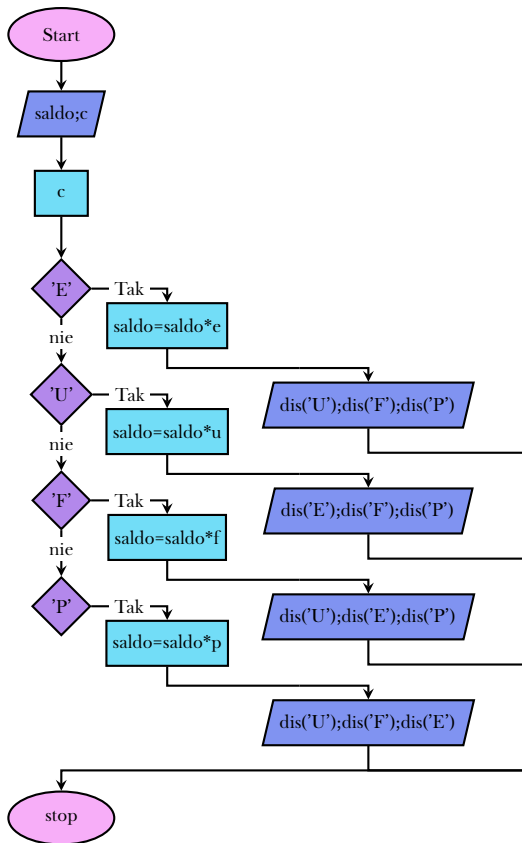


Figure 1: 2.1 flowchart

D:

Saldo, kwota pieniędzy $\in \mathbb{R}$

c, waluta $\in \{u,f,p,e\}$

W: strumień, saldo w innych walutach $\in \mathbb{R}$

```

1  int e=4;int u=3;int f=5;int p=1;int saldo;
2  int main()
3  char c;
4  cout<<"ile masz pieniędzy?:";
5  cin>>saldo;
6  cout<<"w jakiej walucie? \n EUR=E\n\
7  USD=U\n CHF=F\n PLN=P";
8  c=toupper(getch());
9  switch (c)
10 {case 'E':
11     saldo=saldo*e;
12     dis('U');dis('F');dis('P');
13     break;
14 case 'U':
15     saldo=saldo*u;
16     dis('E');dis('F');dis('P');
17     break;
18 case 'F':
19     saldo=saldo*f;
20     dis('U');dis('E');dis('P');
21     break;
22 case 'P':
23     saldo=saldo*p;
24     dis('U');dis('F');dis('E');
25     break;
26 }

```

2.2

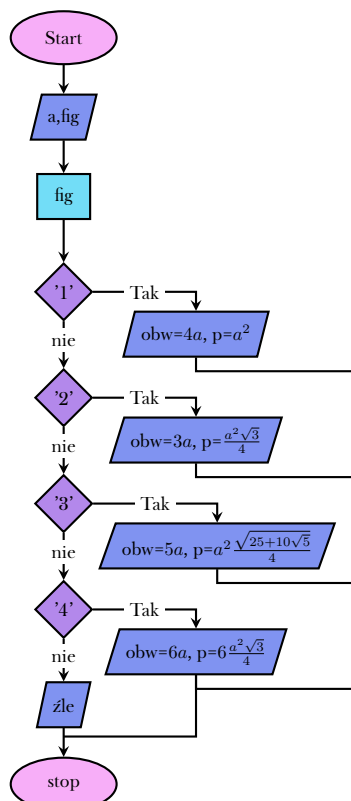


Figure 2: 2.2 flowchart

D: a, długość boku $\geq 0 \in \mathbb{R}$

fig, typ figury $\in \{1;2;3;4\}$

W: p, pole $\geq 0 \in \mathbb{R}$

obw, obwód $\geq 0 \in \mathbb{R}$

```

1  int a;char fig;
2  cout<<"podaj długość boku:"<<endl;
3  cin>>a;cout<<"dostępne figury:\n1.kwadrat\
4  \n2.trójkąt\n3.pięciokąt\n4.sześciokąt";
5  fig=getch();
6  switch (fig)
7  {case '1':
8  cout<<"\nobwód jest równy "<<a*4<<"\
9  a pole "<<a*a;break;
10 case '2':
11 cout<<"\nobwód jest równy "<<a*3<<"\
12 a pole "<<a*a*sqrt(3)/4;break;
13 case '3':
14 cout<<"\nobwód jest równy "<<a*5<<"\
15 a pole "<<(sqrt(25+10*sqrt(5))/4)*a*a;break;
16 case '4':
17 cout<<"\nobwód jest równy "<<a*6<<"\
18 a pole "<<(a*a*sqrt(3)/4)*6;break;
19 default:
20 cout<<"podana opcja jest zła";break;}

```

2.3

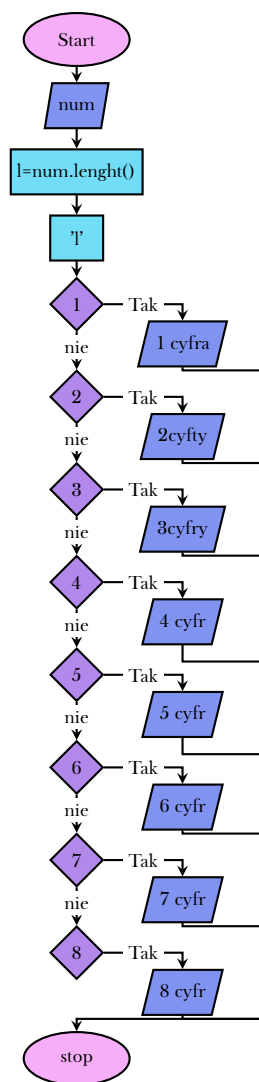


Figure 3: 2.3 flowchart

D: num, ciąg cyfra $\in \mathbb{N}_0$
W: l, liczba cyfr $\in \mathbb{N}_0$

```

1  int l; cout<<"podaj numer:";
2  string num; cin>>num; l=num.length();
3  switch (l)
4  {
5  case 1: {cout<<"numer ma tylko jedną cyfre"; break;}
6  case 2: {cout<<"numer ma dwie cyfry"; break;}
7  case 3: {cout<<"numer ma trzy cyfry"; break;}
8  case 4: {cout<<"numer ma cztery cyfry"; break;}
9  case 5: {cout<<"numer ma pięć cyfr"; break;}
10 case 6: {cout<<"numer ma sześć cyfr"; break;}
11 case 7: {cout<<"numer ma siedem cyfr"; break;}
12 case 8: {cout<<"numer ma osiem cyfr"; break;}
13 case 9: {cout<<"numer ma dziewięć cyfr"; break;}
14 case 10: {cout<<"numer ma dziesięć cyfr"; break;}
15 case 11: {cout<<"numer ma jedynaście cyfr"; break;}
16 default: {cout<<"numer ma "<< l << " cyfr";}
17 }

```

2.4

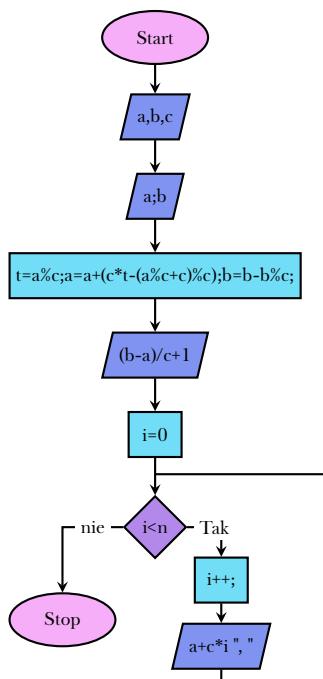


Figure 4: 2.4 flowchart

D: a, dolny limit zakresu $\in \mathbb{Z}$
b, górny limit zakresu $>a \in \mathbb{Z}$
c, dzielnik $\in \mathbb{Z}$

W: strumień, liczby z przedziału podzielne przez $c \in \mathbb{Z}$

```

1  cout<<"ile jest liczb podzielnych\
2  przez c w przedziale a-b"<<endl;
3  cout<<"podaj a:"; int a=0; cin>>a;
4  cout<<"podaj b:"; int b=0; cin>>b;
5  cout<<"podaj c:"; int c=0; cin>>c;
6  cout<<"przdział "<<a<<" ; "<<b<<" zawiera ";
7  bool t=a%c;
8  a=a+(c*t-(a%c+c)%c);
9  b=b-(b%c+c)%c;
10 int n=(b-a)/c+1;
11 cout<<n<<" liczb podzielnych przez "<<c<<endl;
12 for(int i=0; i<n; i++){cout<<a+c*i<<" ";}

```

2.5

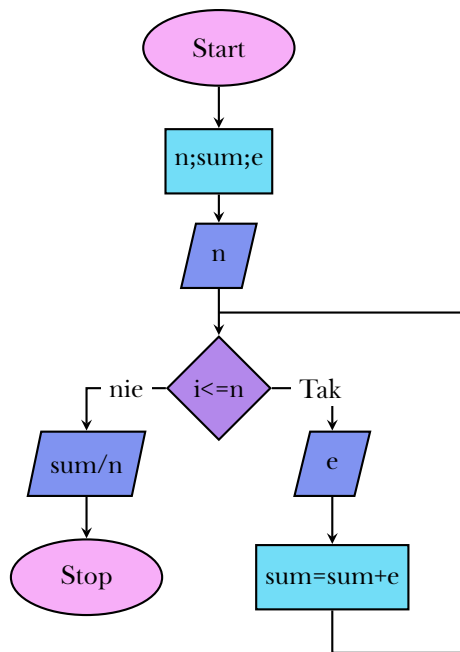


Figure 5: 2.5 flowchart

D:

n , ilość liczb $\in \mathbb{N}$

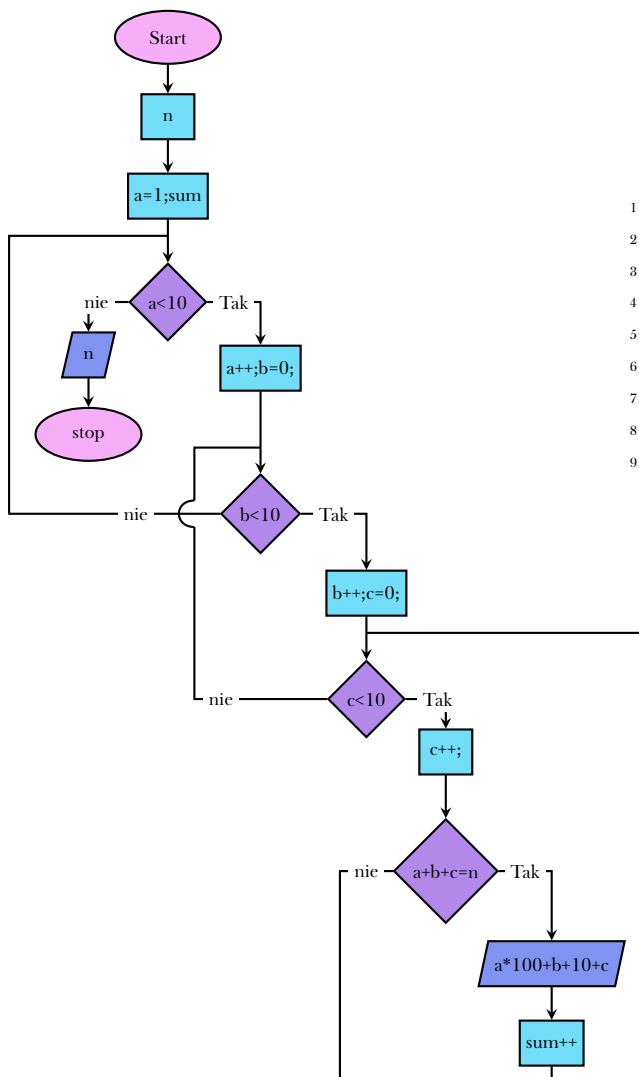
e , n'ta liczba ciągu $\in \mathbb{R}$

W: strumień, średnia ciągu $\in \mathbb{R}$

```

1  cout<<"program do liczenia średniej z n liczb"<<endl;
2  cout<<"podaj n: ";int n=0;cin>>n;
3  float sum=0;float e=0;
4  for(int i=1;i<=n;i++)
5  {cout<<"podaj "<<i<<" liczbę: ";
6   cin>>e;sum=sum+e;}
7  cout<<setprecision(2)<<"średnia "<<n<<"\
8  liczb jest równa: "<<sum/n<<endl;
  
```

2.6



D: n , cyfra $\in \{1,2,3...27\}$

W: strumień, liczby spełniające warunek

i ich ilość $\in \{100,101,102...999\}$

```

1  int n=0;cout<<"podaj n: ";cin>>n;
2  int l[3]= {0,0,0};int p=100;
3  for (int m=round(pow(75.8531-cos(-0.896604*n),\
4   cos(0.0903642*n-1.26534))-4.43137);m>0;p++)
5  {l[0]=floor(p/100);
6   l[1]=floor((p-l[0]*100)/10);
7   l[2]=p-(l[1]*10+l[0]*100);
8   if(l[0]+l[1]+l[2]==n){m--;cout<<p<<endl;}
9  }cout<<"liczbe możliwości: "<<i;
  
```

2.7

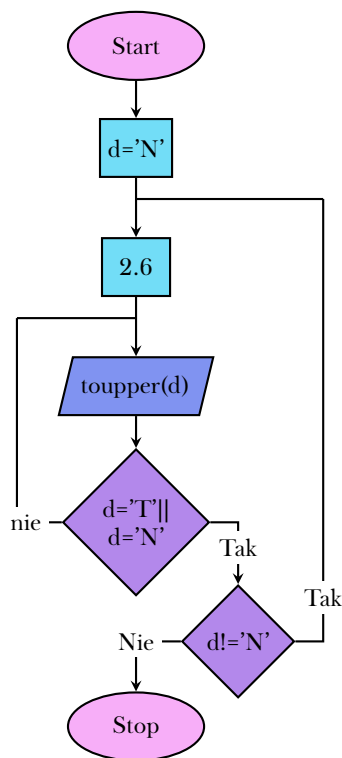


Figure 7: 2.7 flowchart

D: n, cyfra $\in \{1,2,3...27\}$
 W: strumień, liczby spełniające wrunek
 i ich ilość $\in \{100,101,102...999\}$

```

1  int sum=0;
2  char d='N';
3  cout<<"program do liczenie liczb trzy\
4  cyfrowych ktorych suma jest rowna n:"<<endl;
5  do{
6  sum=0;
7  cout<<"podaj n: ";int n;cin>>n;
8  for (int a = 1; a < 10; a++)
9  {for (int b = 0; b < 10; b++)
10 {for (int c = 0; c < 10; c++){
11 if(a+b+c==n){cout<<a*100+b*10+c<<" ";sum++;}
12 }}}
13 cout<<"liczb trzy cyfrowych o sumie "\
14 <<n<<" jest: "<<sum<<endl;
15 cout<<"jeszcze raz? T/N: ";
16 do{
17 cin.clear();cin.sync();
18 cin>>d;
19 d=toupper(d);
20 if(d=='T' || d=='N'){break;}
21 else{cout<<"podana wartosc jest nie poprawna.\
22 Ponow prob" <<endl;}
23 }while(true);
24 }while(d!='N');
  
```

2.8

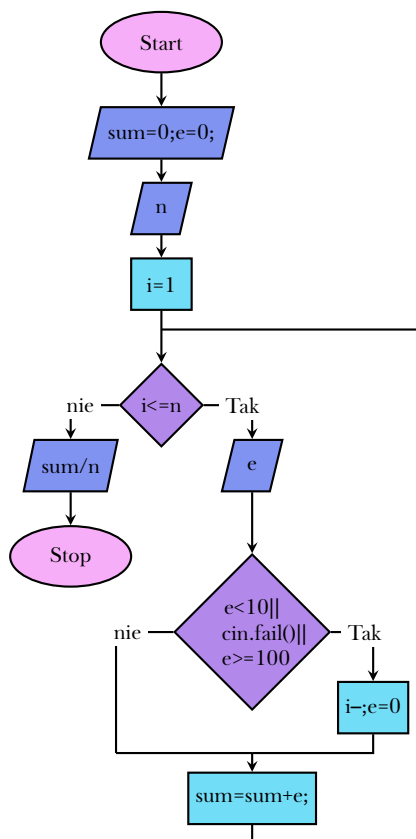


Figure 8: 2.8 flowchart

D:
 n, ilość liczb $\in \mathbb{N}$
 e, n'ta liczba ciągu $\in \{10,11,12...99\}$
 W: strumień, średnia ciągu $\in \mathbb{R}$

```

1  cout<<"program do liczenia &gt;redniaj z\
2  n liczb dwucyfrowych, dodatnich"<<endl;
3  float sum=0;float e=0;
4  cout<<"podaj n: ";int n=0;cin>>n;
5  for(int i=1;i<=n;i++)
6  {cin.clear();cin.sync();
7  cout<<"podaj "<<i<<" liczbe: ";
8  cin>>e;
9  if (cin.fail() || e<10 || e>=100)
10 {cout<<"podana wartosc jest nie poprawna\
11 sprubuj ponownie"<<endl;i--;e=0;}
12 sum=sum+e;}
13 cout<<setprecision(2)<<"srednia "<<n<<"\
14 dwucyfrowych liczb jest rowna: "<<sum/n<<endl;
  
```

2.9

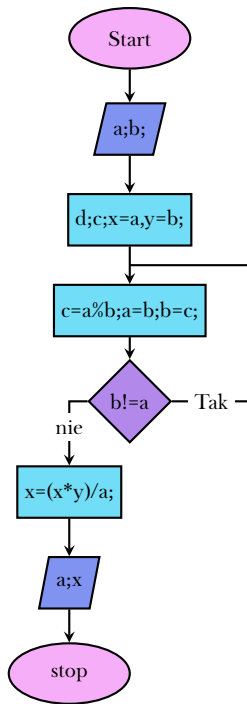


Figure 9: 2.9 flowchart

D: a i b, liczby do nww i nwd $\in \mathbb{N}$
 W: a i x, nww i nwd $\in \mathbb{N}$

```

1  cout<<"program do nwd i nww"<<endl;
2  int i;char l='a';int g[2]={0,0};i=0;
3  while(i<2)
4  {
5      cout<<"podaj "<<l<<": ";
6      cin.clear();cin.sync();
7      g[i]=0;cin>>g[i];i++;l++;
8      if(cin.fail())
9          {cout<<endl<<"podana wartość jest\
10         nie poprawna"<<endl;i--;l--;}
11  }
12  int a=g[0];int b=g[1];
13  int c;int x=a;int y=b;
14  do{c=a%b;a=b;b=c;}while (b!=0);
15  x=(x*y)/a;
16  cout<<"nwd:"<<a<<" nww:"<<x<<endl;
  
```

2.10

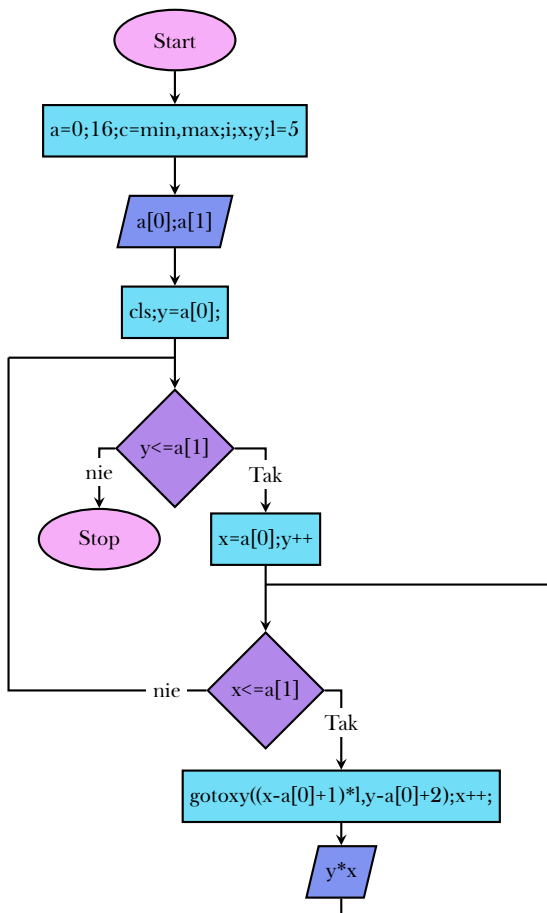


Figure 10: 2.10 flowchart

D: a[], przedział tabliczki mnożenia $\in \{-15,-14,-13...15\}$
 W: strumień, tabliczka mnożenia od a[0] do a[1] $\in \{-225,-224,-223...225\}$

```

1  int a[2]={0,16};
2  string c[2]={"min","max"};
3  int i;int x;int y;int l=4;
4  cout<<"program tabliczka mnozenie od do.\n\
5  Program przyjmuje wartości <-15;15>"<<endl;
6  while(i<2){
7      cout<<"podaj "<<c[i]<<": ";
8      cin.clear();cin.sync();
9      a[i]=0;cin>>a[i];
10     if(cin.fail()||abs(a[i])>15||a[0]>a[1])
11         {cout<<endl<<"podana wartość jest nie poprawna"<<endl;
12         i--;}i++;}system("cls");y=a[0];
13     while (y<=a[1])
14     {
15         x=a[0];cout<<y;gotoxy(3,y-a[0]+2);
16         cout<<"|";
17         while (x<=a[1]){
18             gotoxy((x-a[0]+1)*1,y-a[0]+2);
19             cout<<y*x;
20             x++;}
21         cout<<endl;y++;
22     }
  
```

2.11

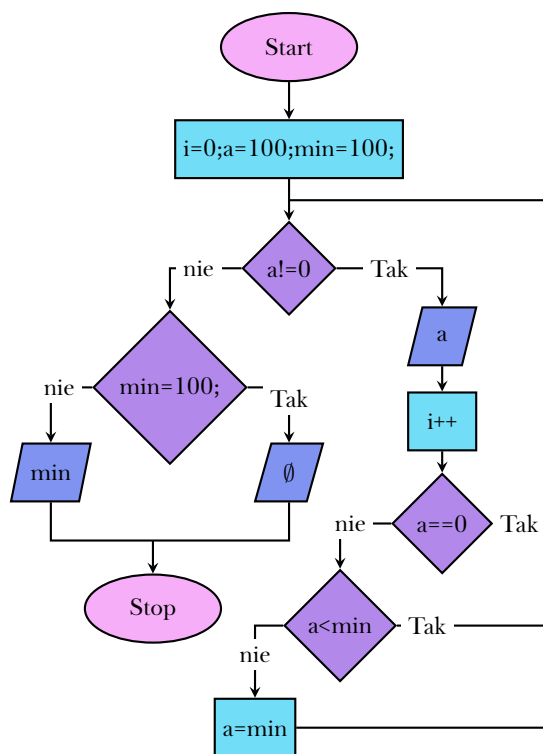


Figure 11: 2.11 flowchart

D: a, n'ta liczba ciągu $\in \mathbb{R}$
 W: min, najmniejszy element ciągu $\in \mathbb{R}$

```

1  int i; int a; int min=100;
2  cout<<"program do szukania minimalne liczby\
3  z ciągu liczb dwu cyfrowych"<<endl;
4  cout<<"aby zakończyć podawanie ciągu\
5  podaj liczbę 0"<<endl;
6  while(a!=0)
7  {
8  i++; cin.clear(); cin.sync();
9  cout<<"podaj "<<i<<": ";
10 a=0; cin>>a;
11 if(a==0){break;}
12 else if(cin.fail() || abs(a)<10 || abs(a)>99)
13 {cout<<"podana wartość jest nie poprawna"\
14 <<endl; i--;}
15 else{if(a<min){min=a;}}
16 }
17 cout<<"minimalną liczbą z ciągu:"<<endl;
18 if(min==100)
19 {cout<<"najmniejsza liczba ciągu należy\
20 do zbioru pustego"<<endl;}
21 else{cout<<endl<<"jest: "<<min<<endl;}
  
```

2.12

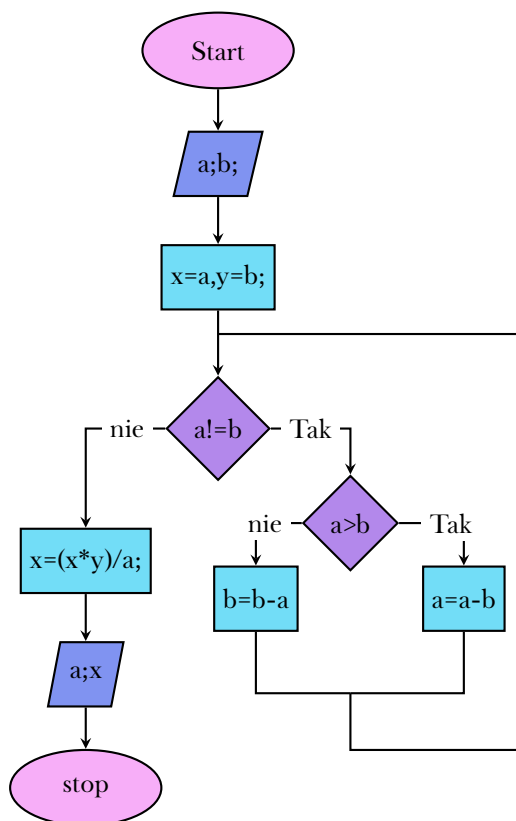


Figure 12: 2.12 flowchart

D: a i b, liczby do nww i nwd $\in \mathbb{N}$
 W: a i x, nww i nwd $\in \mathbb{N}$

```

1  cout<<"program do nwd i nww"<<endl;
2  int i;
3  char l='a';
4  int g[2]={0,0};
5  while(i<2){
6  cout<<"podaj "<<l<<": ";
7  cin.clear(); cin.sync();
8  g[i]=0; cin>>g[i];
9  if(cin.fail())
10 {cout<<endl<<"podana wartość jest nie poprawna"\
11 <<endl; i--; l--;}
12 i++; l++;}
13 int a=abs(g[0]); int x=a;
14 int b=abs(g[1]); int y=b;
15 while(a!=b)
16 {
17     if(a>b){a=a-b;}
18     else{b=b-a;}
19 }
20 x=(x*y)/a;
21 cout<<"nwd: "<<a<< " nww: "<<x<<endl;
  
```

2.13

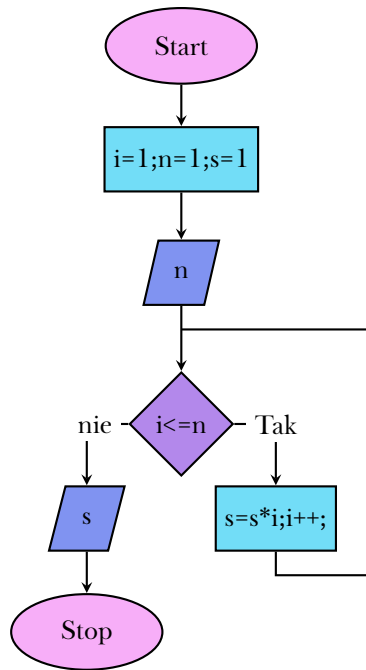


Figure 13: 2.13 flowchart

D: n, liczba do silni $\in \mathbb{N}$
 W: s, silnia n $\in \mathbb{N}$

```

1  Zarezerwuj miejsce dla i, n, s
2  i, n, s równają się 1
3  Zapytaj użytkownika o n
4  s jest równe iloczynowi i oraz s
5  zwiększ i o jeden
6  jeżeli i jest nie mniejsze od n wróć do pkt.4
7  przełącz s do strumienia
  
```

```

1  cout<<"program liczenia silni z n"<<endl;
2  int i;int n;long long s;
3  i=1;n=1;s=1;
4  cout<<"podaj n:";
5  cin.clear();cin.sync();cin>>n;
6  while(i<=n){s=s*i;i++;}
7  cout<<"silna z n jest równa:"<<s<<endl;
  
```

2.14

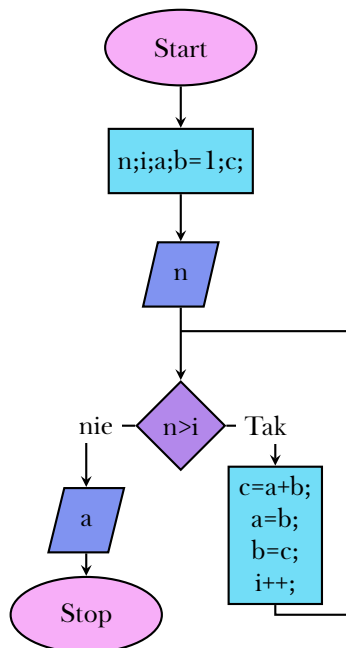


Figure 14: 2.14 flowchart

D: n, index liczby fibonaciego $\in \mathbb{N}$
 W: s, n'ta liczba ciągu fibonaciego $\in \mathbb{N}$

```

1  Zarezerwuj miejsce dla n, i, a, b, c
2  b równa się 1
3  c jest równe sumie a i b
4  a równa się b
5  b równa się c
6  zwiększ i o jeden
7  jeżeli n jest większe od i to wróć do 3 kroku
8  przełącz a do strumienia
  
```

```

1  cout<<endl<<"program do liczenie n liczby fibonaciego"<<endl;
2  cout<<"\npodaj n:";
3  int n = 0;
4  cin>>n;
5  if(cin.fail()){r=false;cout<<endl<<"podana wartość jest nie poprawna"<<endl;}
6  int i = 0;
7  double a=0;double b=1;double c=0;
8  while (n>i)
9  {c=a+b;a=b;b=c;i++;}
10 cout<<n<<" liczba ciągu fibonaciego jest rowna:"<<a<<endl;
  
```


2.15

D: n , liczba do sprawdzenia $\in \mathbb{N}$

W: odp, czy liczba jest pierwsza $\in \{ "tak"; "nie" \}$

```

1  Odp równa się „tak”, i równa się 3 a n równa się jeden
2  Zapytaj użytkownika o n
3  Jeżeli n jest podzielne przez 2 to odp przyjmuje wartość „nie”
4  Jeżeli reszta n/i jest 0 to odp przyjmuje wartość „nie”
5  Zwiększ i o 2
6  Jeżeli i jest mniejszy od pierwiastku z n to idź do czwartego kroku
7  Przekaż odp do strumienia

```

```

1  cout<<"program do sprawdzania czy liczba jest pierwsza"<<endl;
2  int i=3;int n=1;string odp="tak";
3  cout<<"podaj n:";
4  cin.clear();cin.sync();cin>>n;
5  if (n%2==0 && n!=2){odp="nie";}
6  while(i<sqrt(n))
7  {
8      if(n%i==0){odp="nie";break;}
9      i=i+2;
10 }
11 cout<<odp<<endl;

```

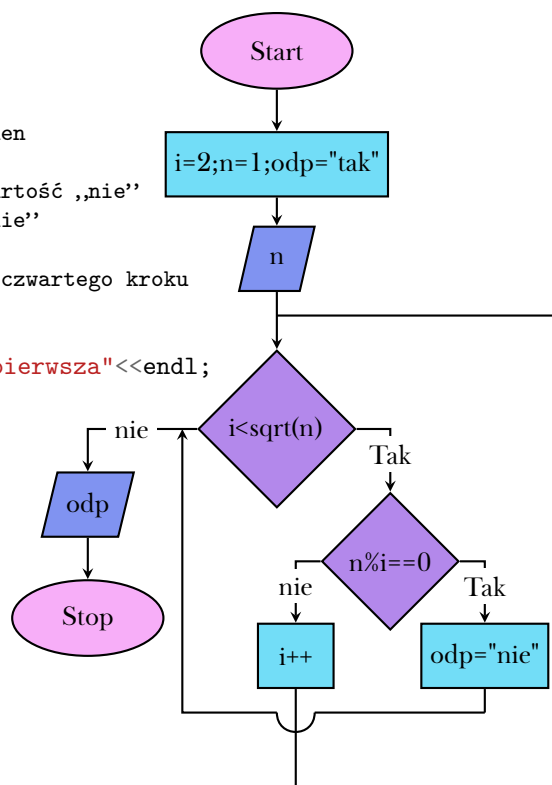


Figure 15: 2.15 flowchart

2.16

D: n , liczba $\in \mathbb{N}_0$

W: s , ilość cyfr w $n \in \mathbb{N}_0$

```

1  Zarezerwuj miejsce dla n i s
2  Zapytaj użytkownika o n
3  Dodaj do s resztę dzielenia dziesiętnego z n
4  Podziel n przez dziesięć
5  Jeżeli n nie równa się zero idź do kroku 3
6  Przekaż s do strumienia

```

```

1  cout<<"program do sprawdzania sumy cyfr n"<<endl;
2  long long int n;
3  int s;
4
5  n=1;s=0;
6  cout<<"podaj n:";
7  cin.clear();cin.sync();cin>>n;
8  while(true)
9  {
10     if(n==0){break;}
11     s=s+n%10;
12     n=n/10;
13 }
14 cout<<"suma wszystkie cyfro to:"<<s<<endl;

```

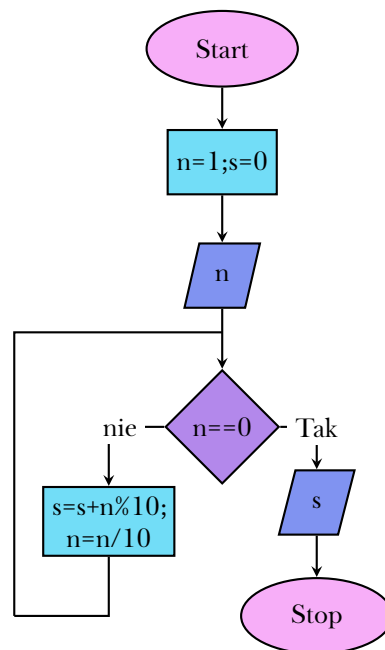


Figure 16: 2.16 flowchart

python

2.1

```

1 import msvcrt
2 def dis(argument):
3     switcher = {
4         'E': print("EUR:",saldo/e,'\n'),
5         'U': print("USD:",saldo/u,'\n'),
6         'F': print("CHF:",saldo/f,'\n'),
7         'P': print("PLN:",saldo/p,'\n'),
8     }
9 e=4;u=3;f=5;p=1
10 saldo=int(input('Ile masz pieniedzy:'))
11 print('w jakiej walucie')
12 c=msvcrt.getche().upper()
13 print(c)
14 match c:
15     case b'E':
16         saldo=saldo*e; dis('E'),
17     case b'U':
18         saldo=saldo*u; dis('U'),
19     case b'F':
20         saldo=saldo*f; dis('F'),
21     case b'P':
22         saldo=saldo*p; dis('P'),

```

2.2

```

1 import msvcrt
2 import math
3 a=int(input('podaj długość boku:'))
4 print('dostępne figury:\n1.kwadrat\n2.trójkąt\n3.pięciokąt\n4.sześciokąt')
5 fig=msvcrt.getche().upper()
6 match fig:
7     case b'1':
8         print("\nobwód jest równy ",a*4," a pole ",a*a)
9     case b'2':
10        print("\nobwód jest równy ",a*3," a pole ",a*a*math.sqrt(3)/4)
11    case b'3':
12        print("\nobwód jest równy ",a*4," a pole ",(math.sqrt(25+10*math.sqrt(5))/4)*a*a)
13    case b'4':
14        print("\nobwód jest równy ",a*5," a pole ",(a*a*math.sqrt(3)/4)*6)
15    case _:
16        print("\nzłą opcja")

```

2.3

```

1 l=int(input('podaj numer:'))
2 l=len(str(l))
3 match l:
4     case 1:print("numer ma jedną cyfre")
5     case 2:print("numer ma dwie cyfry")
6     case 3:print("numer ma trzy cyfry")
7     case 4:print("numer ma cztery cyfry")
8     case 5:print("numer ma pięć cyfr")
9     case 6:print("numer ma sześć cyfr")
10    case 7:print("numer ma siedem cyfr")
11    case 8:print("numer ma osiem cyfr")
12    case 9:print("numer ma dziewięć cyfr")
13    case 10:print("numer ma dziesięć cyfr")
14    case 11:print("numer ma jedynaście cyfr")
15    case _:print('numer ma ',l," cyfr")

```

2.4

```

1 print('ile jest liczb podzielnych przez c w <a;b>')
2 a=int(input('podaj a:'))
3 b=int(input('podaj b:'))
4 c=int(input('podaj c:'))
5 t=bool(a%c)
6 a=a+(c*t-a%c)
7 b=b-b%c
8 n=int((b-a)/c+1)
9 print(n,' liczb podzielnych przez ',c)
10 for i in range(0,n):print(a+c*i,', ',end="")

```

2.5

```

1 print('program do liczenia średniej z n liczb')
2 n=int(input('podaj n:'))
3 sum=float(0)
4 for i in range(0,n):
5     print('podaj ',i+1,', ',end="")
6     e=int(input())
7     sum=sum+e
8 sum=round(sum/n,2)
9 print('średnia',n,'liczba jest równa: ',sum)

```

2.6

```

1 print('szuka liczb 3 cyfrowych ktorych suma jest rowna n:')
2 n=int(input('podaj n:'))
3 sum=int(0)
4 for a in range(1,10):
5     for b in range(0,10):
6         for c in range(0,10):
7             if a+b+c==n:
8                 print(a*100+b*10+c," ",end='')
9                 sum=sum+1
10 print("\nliczb trzy cyfrowych o sumie ",n," jest: ",sum)

```

2.7

```
1 import msvcrt as m
2 import sys
3 print('program do liczenie liczb trzy cyfrowych ktorych suma jest rowna n:')
4 while True:
5     n=int(input('podaj n:'))
6     sum=int(0)
7     for a in range(1,10):
8         for b in range(0,10):
9             for c in range(0,10):
10                 if a+b+c==n:print(a*100+b*10+c," ",end='');sum=sum+1
11 print("\nliczb trzy czyfrowych o sumie ",n," jest: ",sum)
12 while True:
13     print('kontynować? T/N')
14     s=m.getche().upper()
15     if s==b'T':break
16     elif s==b'N':sys.exit(0)
```

2.8

```
1 print('program do liczenina &gt;redniaj z n liczb dwucyfrowych, dodatnich')
2 n=int(input('podaj n:'))
3 sum=float(0)
4 e=int(0)
5 i=int(1)
6 while i <= n:
7     try:
8         print('podaj',i,'liczbe:')
9         e=int(input())
10        if e<10 or e>=100:
11            i=i-1
12            print('poza zakresem :(')
13        else:sum=sum+e
14    except ValueError:
15        print('zła wartość')
16        i=i-1
17        i=i+1
18 sum=round(sum/n,2)
19 print('średnia jest równa: ',sum)
```

2.9

```
1 print('program do nwd i nww')
2 a=int(0)
3 b=int(0)
4 i=int(1)
5 while True:
6     try:
7         a=int(input('podaj a liczbe:'))
8         break
9     except ValueError:print('zła wartość')
10 while True:
11     try:
12         b=int(input('podaj b liczbe:'))
13         break
14     except ValueError:print('zła wartość')
15 c=int(0)
16 x=a
17 y=b
18 print('test')
19 while b != 0:
20     c = a % b
21     a = b
22     b = c
23     print('test')
24 x=int((x*y)/a)
25 print("nwd:", a, "nww:", x)
```

2.10

```
1 k = b'n'
2 min,max = 0,0
3 while True:
4     try:
5         min = int(input("podaj min:"))
6         if -15<=min and min<=15:
7             break
8         else:print("Poza zakresem")
9     except ValueError:print("Zła wartość")
10 while True:
11     try:
12         max = int(input("podaj max:"))
13         if -15<=max and max<=15 :
14             if max<min:
15                 print("min musi byc mniejsze od max")
16             else:
17                 break
18         else:
19             print("Poza zakresem")
20     except ValueError:
21         print("Zła wartość")
22 l1,l3 = min,min
23 print("      ",end="")
24 while l3 <=max:
25     print(f"{l3:>4}",end="")
26     l3 += 1
27 print("")
28 while l1 <=max:
29     print(f"{l1:<4}",end="")
30     l2 = min
31     while l2 <=max:
32         print(f"{l1*l2:>4}",end="")
33         l2 += 1
34     print("")
35     l1 += 1
```

2.11

```
1 print("program do szukania minimalne liczby z ciągu liczb dwu cyfrowych")
2 print("aby zakończyć podawanie ciągu podaj liczbę 0")
3 i = 0
4 a=1
5 min_val = 100
6 while a != 0:
7
8     i += 1
9     while True:
10         try:
11             a=int(input('podaj liczbę:'))
12             if abs(a) > 10 and abs(a) < 99:break
13             if a==0:break
14             print("podana wartość jest nie poprawna")
15         except ValueError:
16             print('zła wartość')
17             i=i-1
18     if a == 0:
19         break
20     elif a < min_val: min_val=a
21 print("minimalną liczbą z ciągu:")
22 if min_val == 100:
23     print("najmniejsza liczba ciągu należy do zbioru pustego")
24 else:
25     print("jest: ", min_val)
```

2.12

```
1 print('program do nwd i nww')
2 a=int(0)
3 b=int(0)
4 i=int(1)
5 while True:
6     try:
7         a=int(input('podaj a liczbe:'))
8         break
9     except ValueError:print('zła wartość')
10 while True:
11     try:
12         b=int(input('podaj b liczbe:'))
13         break
14     except ValueError:print('zła wartość')
15
16 x=a
17 y=b
18 while a!=b:
19     if a>b:
20         a=a-b
21     else :b=b-a
22 x=int((x*y)/a)
23 print("nwd:", a, "nww:", x)
```

2.13

```
1 print('program liczenia silni z n')
2 s=int(1)
3 n=int(1)
4 i=int(1)
5 while True:
6     try:
7         n=round(int(input('podaj a liczbe:')))
8         break
9     except ValueError:print('zła wartość')
10
11 while i<=n:
12     s=s*i
13     i=i+1
14 print("silna z n jest równa:",s)
```

2.14

```
1 print('program do liczenie n liczby fibonacciego')
2 i=int(0)
3 n=int(1)
4 a=int(0)
5 b=int(1)
6 c=int(0)
7 while True:
8     try:
9         n=round(int(input('podaj a liczbe:')))
10        break
11    except ValueError:print('zła wartość')
12 while n>i:
13     c=a+b
14     a=b
15     b=c
16     i=i+1
17 print(n," liczba ciagu fibonacciego jest rowna:",a)
```

2.15

```
1 import math
2 print('sprawdzanie czy liczba jest pierwsza')
3 i=int(2)
4 n=int(1)
5 odp='tak'
6 while True:
7     try:
8         n=int(input('podaj a liczbe:'))
9         break
10    except ValueError:print('zła wartość')
11
12 while i<=math.sqrt(n):
13     if n%i==0:
14         odp='nie'
15         break
16     i=i+1
17 print(odp)
```

2.16

```
1 import math
2
3 print('program do sprawdzania jaka jest suma cyfr n')
4 s=int(0)
5 n=int(1)
6 while True:
7     try:
8         n=round(int(input('podaj a liczbe:')))
9         break
10    except ValueError:print('zła wartość')
11
12 while True:
13     if n==0:
14         break
15     s=s+n%10
16     n=round(n/10)
17 print('Suma wszystkich cyfr to', s)
```

Programy użyte do wykonania zadań

LaTeX, tikz, firefox, overleaf, dev c++, c++, python, visual studio code, notepad++, git, github, Sumatra PDF, Total commander

Wnioski i uwagi

Zadanie mi się bardzo podobało i nie mam żadnych uwag.