Zadanie 2.22. Podaj specyfikację zadania i skonstruuj algorytm w postaci programu wykonujący dodawanie dwóch wprowadzonych z klawiatury nieujemnych liczb całkowitych zapisanych w systemie liczbowym o podstawie z przedziału [2, 9], również wprowadzonej z klawiatury. Wynik niech będzie wypisany w tym samym systemie.

2.3.4. Wyznaczanie wartości wielomianu za pomocą schematu Hornera

Schemat Hornera jest najszybszym sposobem obliczania wartości wielomianu. Przeanalizujmy działanie tej metody, przekształcając ogólny wzór na wartość wielomianu stopnia n.

Dany mamy wielomian stopnia n, gdzie $n \geq 0$:

$$w_n(x) = a_0x^n + a_1x^{n-1} + \ldots + a_{n-1}x + a_n\,. \tag{2.10}$$

W omawianym algorytmie należy stosować grupowanie wyrazów tak długo, aż pozostanie dwumian.

$$\begin{aligned} w_n(x) &= a_0x^n + a_1x^{n-1} + \ldots + a_{n-1}x + a_n = \\ &= \left(a_0x^{n-1} + a_1x^{n-2} + \ldots + a_{n-1}\right)x + a_n = \\ &= \left(\left(a_0x^{n-2} + a_1x^{n-3} + \ldots + a_{n-2}\right)x + a_{n-1}\right)x + a_n = \\ &= \ldots = \\ &= \left(\left(\ldots\left(\left(a_0x + a_1\right)x + a_2\right)x + \ldots + a_{n-2}\right)x + a_{n-1}\right)x + a_n \end{aligned} \tag{2.11}$$

Schemat Hornera ma więc następującą postać:

$$w_n(x) = \left(\left(\ldots\left(\left(a_0x + a_1\right)x + a_2\right)x + \ldots + a_{n-2}\right)x + a_{n-1}\right)x + a_n\,. \tag{2.12}$$

Porównując wzory 2.10 i 2.12 na obliczanie wartości wielomianu, łatwo zauważyć, że w schemacie Hornera wykonywana jest mniejsza liczba mnożeń.

Przykład 2.17.

Obliczmy wartość wielomianu $w(x) = 2x^3+4x^2-3x+7$ dla $x = 3$, wykorzystując schemat Hornera. Współczynnikami wielomianu są tutaj $a_0 = 2$, $a_1 = 4$, $a_2 = -3$, $a_3 = 7$, a stopień wielomianu n wynosi 3.

$$\begin{aligned} w(x) &= 2x^3 + 4x^2 - 3x + 7 = \\ &= \left(\left(2x + 4\right)x - 3\right)x + 7 \end{aligned}$$

Stąd dla $x = 3$ mamy:

$$\begin{aligned} w(3) &= 2 \cdot 3^3 + 4 \cdot 3^2 - 3 \cdot 3 + 7 = \\ &= \left(\left(2 \cdot 3 + 4\right) \cdot 3 - 3\right) \cdot 3 + 7 = \\ &= \left(10 \cdot 3 - 3\right) \cdot 3 + 7 = \\ &= 27 \cdot 3 + 7 = \\ &= 88 \end{aligned}$$

Porównajmy liczbę działań wykonywanych przy obliczaniu wartości wielomianu po wybraniu każdego ze wzorów.

$$w(x) = 2 \cdot x \cdot x \cdot x + 4 \cdot x \cdot x + (-3) \cdot x + 7$$

W powyższym przykładzie wykonano sześć operacji mnożenia oraz trzy operacje dodawania.

W przypadku schematu Hornera wzór można przedstawić następująco:

$$w(x) = ((2 \cdot x + 4) \cdot x + (-3)) \cdot x + 7.$$

Liczba wykonanych działań jest tutaj znacznie mniejsza: trzy mnożenia i trzy dodawania.

Zadanie 2.23. Opierając się na powyższej analizie, wyznacz ogólną liczbę operacji mnożenia i dodawania przy obliczaniu wartości wielomianu stopnia n. Na podstawie uzyskanych wyników podaj złożoność czasową obydwu algorytmów.

Wyznaczając wartość wielomianu schematem Hornera (patrz wzór 2.12), należy wykonać następujące operacje:

$$\begin{aligned} w &= a_0 \\ w &= wx + a_1 \\ w &= wx + a_2 \\ w &= wx + a_3 \\ &\dots \\ w &= wx + a_{n-1} \\ w &= wx + a_n \end{aligned} \qquad (2.13)$$

Możemy więc zdefiniować **wzór iteracyjny** tego algorytmu:

$$\begin{cases} w = a_0 \\ w = wx + a_i \quad dla \quad i = 1, 2, \dots, n \end{cases} \qquad (2.14)$$

Na podstawie otrzymanego wzoru 2.14 konstruujemy **algorytm iteracyjny** w postaci listy kroków, schematu blokowego (patrz rysunek 2.6) oraz programów w językach C++ i Pascal.

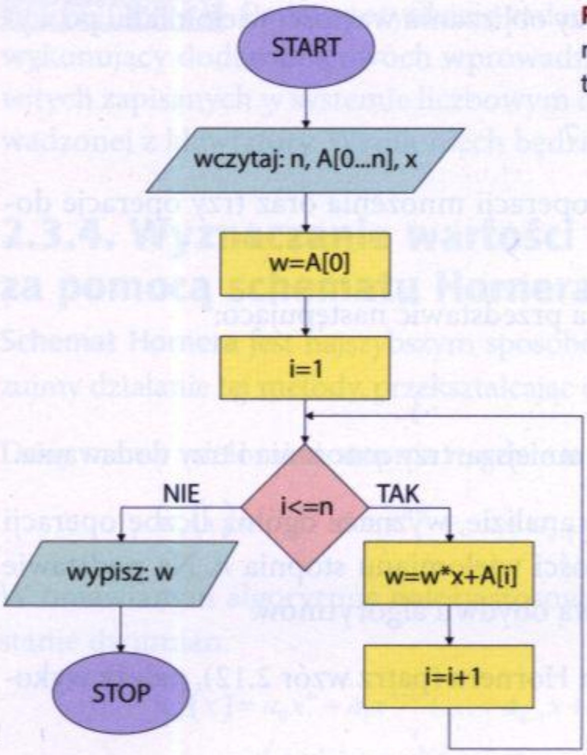
Specyfikacja:

Dane: Liczba całkowita: $n \geq 0$ (stopień wielomianu).

$n+1$-elementowa tablica liczb rzeczywistych: $A[0...n]$ (współczynniki wielomianu).

Liczba rzeczywista: x (wartość argumentu).

Wynik: Wartość rzeczywista wielomianu stopnia n dla wartości argumentu x.



Rysunek 2.6. Schemat blokowy algorytmu iteracyjnego wyznaczającego wartość wielomianu schematem Hornera

Lista kroków:

Krok 0. Wczytaj wartości danych n, $A[0...n]$, x.

Krok 1. Przypisz $w = A[0]$.

Krok 2. Dla kolejnych wartości i: 1, 2, …, n, wykonuj krok 3.

Krok 3. Przypisz $w = w \cdot x + A[i]$.

Krok 4. Wypisz wartość wielomianu: w. Zakończ algorytm.

Funkcja w języku C++ *(prog2_9.cpp)****:***

```
double oblicz (double A[], int n, double x)
{
  double w=A[0];
  for (int i=1;i<=n;i++) w=w*x+A[i];
  return w;
}
```

Funkcja w języku Pascal *(prog2_9.pas)****:***

```
function oblicz (A: tablica; n: integer; x: real): real;
var w: real;
    i: integer;
begin
  w:=A[0];
```

```
for i:=1 to n do w:=w*x+A[i];
oblicz:=w
end;
```

Przedstawiony algorytm można wykonać również rekurencyjnie. Nietrudno zauważyć zależność rekurencyjną, na podstawie której obliczana jest wartość wielomianu stopnia n.

$$w_n(x) = a_0x^n + a_1x^{n-1} + \ldots + a_{n-1}x + a_n = \underbrace{\left(a_0x^{n-1} + a_1x^{n-2} + \ldots + a_{n-1}\right)}_{w_{n-1}(x)}x + a_n = w_{n-1}(x)x + a_n \quad \textbf{(2.15)}$$

Na podstawie wzoru 2.15 tworzymy **definicję rekurencyjną**, która wygląda następująco:

$$w_n(x) = \begin{cases} a_0 & dla \quad n = 0 \\ w_{n-1}(x) \cdot x + a_n & dla \quad n > 0 \end{cases} \quad \textbf{(2.16)}$$

Wskazówka

Zastosowanie schematu Hornera nie ogranicza się do wyznaczania wartości wielomianu stopnia n. Algorytm ten wykorzystywany jest również do:

- konwersji liczb z dowolnego pozycyjnego systemu liczbowego na system dziesiętny;
- szybkiego obliczania wartości potęgi;
- jednoczesnego obliczania wartości wielomianu i jego pochodnej.

Zadanie 2.24. Napisz program obliczający rekurencyjnie wartość wielomianu stopnia n z wykorzystaniem schematu Hornera, zgodny z podaną powyżej specyfikacją algorytmu iteracyjnego.

2.3.5. Zamiana liczb z dowolnego pozycyjnego systemu liczbowego na system dziesiętny z zastosowaniem schematu Hornera

Schemat Hornera można zastosować do konwersji liczb zapisanych w różnych systemach liczbowych na system dziesiętny. Przypomnijmy, w jaki sposób dokonujemy takiej zamiany w systemie binarnym, co zostało omówione w punkcie 2.3.2, „Konwersje pozycyjnych systemów liczbowych". Zamieńmy liczbę 1011101_2 na wartość w systemie decymalnym:

$$1011101_2 = 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 93_{10}.$$

Łatwo zauważyć, że zapis liczby podczas obliczeń przypomina wielomian. Cyfry zamienianej liczby można więc potraktować jak współczynniki wielomianu, a podstawę systemu jak wartość argumentu x. W tym przypadku mamy następującą sytuację:

$n = 6,$

$a_0 = 1,$

$a_1 = 0,$

$a_2 = 1,$

$a_3 = 1,$

$a_4 = 1,$

$a_5 = 0,$

$a_6 = 1,$

$x = 2.$

Taka interpretacja konwersji liczb na system dziesiętny umożliwia zastosowanie do jej realizacji schematu Hornera. Skonstruujmy więc **algorytm wykonujący zamianę liczb z systemu dwójkowego na dziesiętny**.

Specyfikacja:

Dane: Liczba całkowita: $n \geq 0$ (stopień wielomianu).

n+1-elementowa tablica liczb całkowitych: $A[0...n]$ (współczynniki wielomianu, czyli cyfry liczby zapisanej w systemie binarnym).

Wynik: Wartość wielomianu stopnia n dla argumentu 2 (liczba w systemie dziesiętnym).

Funkcja w języku C++ *(prog2_10.cpp):*

```
long oblicz (int A[], int n)
{
 long w=A[0];
 for (int i=1;i<=n;i++) w=w*2+A[i];
 return w;
}
```

Funkcja w języku Pascal *(prog2_10.pas):*

```
function oblicz (A: tablica; n: integer): longint;
var w: longint;
    i: integer;
begin
 w:=A[0];
 for i:=1 to n do w:=w*2+A[i];
 oblicz:=w
end;
```

Zadanie 2.25. Podaj specyfikację zadania i skonstruuj rekurencyjny algorytm w postaci programu realizujący konwersję liczb zapisanych w systemie o podstawie p, gdzie p jest dowolną liczbą naturalną z przedziału [2, 9], na system dziesiętny z zastosowaniem schematu Hornera.

Zadanie 2.26. Podaj specyfikację zadania i skonstruuj iteracyjny algorytm w postaci programu realizujący konwersję liczb zapisanych w systemie szesnastkowym na system dziesiętny z zastosowaniem schematu Hornera.