

Programowanie

W praktyce inżynierskiej zdarza się, że nie mamy możliwości opisać zagadnienia w postaci jednego wyrażenia algebraicznego. Na przykład należy wybrać spośród kilku możliwości (np. wartości ujemne - jeden wzór, wartości dodatnie - drugi wzór), albo też wiele razy powtórzyć tę samą operację (np. dodawaj do poprzedniego wyrażenia).

Z pomocą przychodzi nam programowanie. Wydawać by się mogło, że jest programowanie, to zadanie niezmiernie trudne i skomplikowane. Jednak dla naszych celów będziemy stosować tylko niewielki wycinek ogromnych możliwości, jakie w sobie kryje program.

Opanowanie tych podstawowych elementów spowoduje, że różnorodne zadania, które będziemy rozwiązywać staną się bardziej zrozumiałe i łatwiejsze - osiągniemy zupełnie inny (wyższy) poziom logicznego rozumowania. A funkcje, które sami napiszemy będziemy mogli wykorzystać w dalszej pracy.

Podstawowe elementy programu

blok LINE - kilka instrukcji naszego programu wykonywanych w jednym ciągu

instrukcja warunkowa IF - rozgałęzienie ścieżki wykonywania naszego programu

instrukcja pętli FOR - powtarzamy instrukcję lub blok instrukcji określoną ilość razy

instrukcja pętli WHILE - powtarzamy instrukcję dopóki warunek jest prawdziwy.

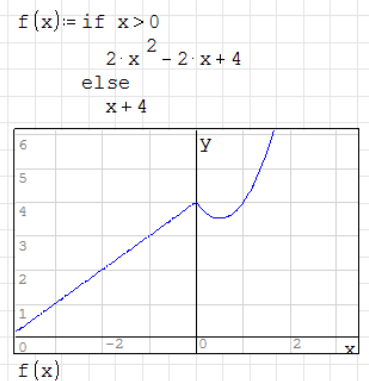
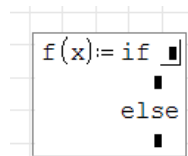
ĆWICZENIE 1 - instrukcja warunkowa

Narysuj wykres funkcji określonej w następujący sposób:

$$f(x) = \begin{cases} 2x^2 - 2x + 4 & x > 0 \\ x + 4 & x \leq 0 \end{cases}$$

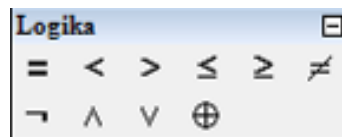
- Wpisz: `f(x → : if TAB`

Pojawi się schemat instrukcji warunkowej. Poszczególne pola wypełniamy jak na rysunku.



Operatory logiczne

Do porównywania różnych wartości służą operatory logiczne, zgrupowane na panelu Logika. Górny wiersz nie wymaga wyjaśnienia. W dolnym znajdują się operacje logiczne złożone omawiane na matematyce. W kolejności: zaprzeczenie, iloczyn logiczny (koniunkcja), suma logiczna (alternatywa) i mniej znana różnica symetryczna.



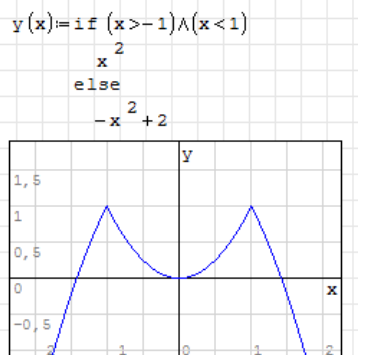
ĆWICZENIE 2 - część wspólna

Narysuj wykres funkcji określonej w $f(x) = \begin{cases} x^2 & -1 < x < 1 \\ -x^2 + 2 & -1 \geq x \geq 1 \end{cases}$ następujący sposób:

Wystarczy poprawnie opisać jeden warunek, cała reszta należy do drugiego przedziału.

Zapis `if (x > -1) ^ (x < 1)` oznacza, że warunek jest prawdziwy dla x-ów należących do przedziału otwartego od -1 do 1, czyli $-1 < x < 1$.

UWAGA - przedział rozłączny określamy za pomocą sumy logicznej, a iloczynem logicznym oznaczamy część wspólną.



ĆWICZENIE 3 - pierwiastki równania kwadratowego

Utwórz funkcję o nazwie PIER z trzema parametrami (a, b, c) – są to współczynniki równania kwadratowego. Funkcja ma zwracać wyliczone pierwiastki, gdy $\Delta \geq 0$ lub napis „brak pierwiastków”, gdy $\Delta < 0$.

Definiujemy funkcję o nazwie PIER z trzema parametrami a, b, c. W zmiennej DELTA wyliczamy deltę. Instrukcja warunkowa: jeżeli DELTA większa lub równa 0, to liczymy dwa pierwiastki (znak ±), w przeciwnym wypadku napis „brak pierwiastków”

```
pier(a ; b ; c) := | delta:=b^2-4*a*c
                  | if delta >= 0
                  |   -b+sqrt(delta)
                  |   -b-sqrt(delta)
                  |   2*a
                  | else
                  |   "brak pierwiastków"
```

```
pier(2 ; 4 ; 1) = { -0,2929
                  | -1,7071
```

ĆWICZENIE 4 - rozdział

Utwórz funkcję o nazwie NIM z jednym parametrem, której zadaniem będzie rozdzielenie nazwiska i imienia podanego, jako parametr na dwa osobne wyrazy. Jeżeli w parametrze nie wystąpi znak odstępu, funkcja nie zwróci nic.

Tworzymy funkcję o nazwie NIM z parametrem „napis”

W zmiennej S pozycja spacji w napisie za pomocą funkcji FINDSTR

Jeżeli zmienna S jest równa -1 (nie znalazło spacji) to wypisz pusty napis „”

W przeciwnym wypadku (znalazło spację) obliczamy długość tekstu w zmiennej D

Za pomocą układu równań tworzymy listę z dwoma fragmentami napisu.

```
nim(napis) := | S:= findstr(napis ; " ")
              | if S=-1
              |   ""
              | else
              |   D:= strlen(napis)
              |   { substr(napis ; 1 ; S-1)
                  |   substr(napis ; S+1 ; D-S)
```

```
nim("wacek libront") = { "wacek"
                       | "libront"
```

```
nim("waceklibront") = ""
```

Pętla FOR

ĆWICZENIE 5 - średnia ocen

Lista zawiera oceny ucznia z pewnego przedmiotu. Napisz funkcję o nazwie ŚRE, która za pomocą instrukcji FOR wyliczy średnią ocen. Jako parametr podajemy listę z ocenami.

Aby wyliczyć średnią arytmetyczną musimy podsumować wszystkie oceny z listy i podzielić przez ich liczbę.

Przyjmujemy, że początkowa suma będzie wynosiła zero (ważne)!

Wyliczamy długość listy - ilość elementów.

W pętli FOR sumujemy wszystkie elementy listy: do sumy dodajemy kolejny element i znów zapisujemy w tej samej zmiennej (tak można)! Mówimy, że powiększamy zmienną suma o i-ty element listy

Przypisujemy do nazwy funkcji iloraz sumy przez ilość elementów.

Przykładowe wywołanie funkcji ŚRED z listą ocen OC i listą liczb parzystych 2..100

```
(1)  sred(lista):= suma:=0
(3)  ile:=length(lista)
(4)  for i∈1..ile
(6)  suma:=suma+lista_i
(5)  sred:=suma/ile

OC:=
(4)
(2)
(5)

sred(OC)=3,75
sred(2;4..100)=51
```

ĆWICZENIE 6 - maksymalna ocena

Utwórz funkcję o nazwie MAX, która wyszuka w liście podanej, jako parametr największą ocenę.

W zmiennej ILE wyliczamy ilość elementów listy.

W zmiennej M (ustawionej na zero) będziemy przechowywać maksymalną wartość.

W pętli FOR, gdzie I zmienia się od 1 do ILE

Sprawdzamy każdy element listy, czy nie jest większy od zmiennej M.

Jeżeli jest większy, to do zmiennej M zapisujemy ten element listy.

W przeciwnym razie nie robimy nic

Po zakończeniu pętli do nazwy funkcji wpisujemy wartość zmiennej M

```
max(lista):= ile:=length(lista)
m:=0
for i∈1..ile
if lista_i>m
m:=lista_i
else
0
max:=m
```

max(OC)=6

ĆWICZENIE 7 - losowe oceny

Wypełnij listę losowymi ocenami

W pętli FOR wpisujemy do elementów listy losową liczbę - funkcja RANDOM.

Random(6) losuje liczby z przedziału 0..5, a random(6)+1 z przedziału 1..6.

Za każdym razem, gdy naciśniemy CTRL+F9 losowanie zostanie powtórzone

```
for i∈1..8
OC_i:=random(6)+1
OC=
(3)
(4)
(5)
(4)
(2)
(5)
```

ĆWICZENIE 8 - arkusz ocen

Utwórz macierz o wymiarach 10x10 i wylosuj do wszystkich jej elementów oceny

Funkcja MATRIX definiuje pustą macierz same zera o wymiarach 10x10.

Podwójna pętla FOR wstawia losowe oceny do każdego elementu tablicy zgodnie z numerem indeksu.

Element tablicy A_{ij} należy wpisać w następujący sposób:

- Wpisz: A [i ; j

Nie można pominąć średnika, który jest niezbędny!

```
A:=matrix(10;10)
for i∈1..10
for j∈1..10
A_i_j:=random(6)+1
A=
(3 6 4 3 1 1 1 2 5 3)
(1 5 4 3 3 1 5 4 6 2)
(5 6 1 1 3 3 3 3 4 5)
(5 6 4 4 3 5 2 3 3 6)
(5 5 5 2 5 4 3 1 6 1)
(4 3 6 4 6 3 5 4 6 4)
(5 4 4 1 4 2 2 1 1 4)
(6 2 2 1 4 1 1 6 3 5)
(1 2 4 3 2 2 1 1 4 6)
(6 5 3 5 2 2 1 5 4 3)
```

ĆWICZENIE 9 - ilość ocen

Policz wszystkie oceny z tablicy A

Tworzymy zerowy wektor OC na ilości poszczególnych ocen.

W elemencie pierwszym OC_1 jedynki, OC_2 dwójki itd.

ROWS i COLS podają nam ilość wierszy i kolumn tablicy

Podwójna pętla pobiera do zmiennej EL po kolei każdą ocenę tablicy.

Ocena jest jednocześnie indeksem z wektora OC.

Więc odpowiednią komórkę wektora OC (ocena) zwiększamy o jeden.

```
ILOC(mac):= oc:=matrix(6;1)
w:=rows(mac)
k:=cols(mac)
for i∈1..w
for j∈1..k
el:=A_i_j
oc_el:=oc_el+1
ILOC:=oc
ILOC(A)=
(16)
(17)
(11)
(18)
(19)
(19)
```

Pętla WHILE

Pętla FOR wykonywała się określoną, konkretną ilość razy. Pętla logiczna WHILE nie będzie miała narzuconej krotności. Za każdym razem sprawdzany jednak będzie warunek zakończenia. Gdy stanie się prawdziwy - pętla skończy działanie. Na przykład szukamy w tablicy pierwszego elementu większego od zera.

ĆWICZENIE 10 - losowe oceny

Wypełnij listę losowymi ocenami.

Zadanie z ćwiczenia 6 wykonamy za pomocą pętli WHILE. Najpierw ustawiamy zmienną I, która „steruje” warunkiem zakończenia pętli. Pętla kończy działanie, gdy zmienna sterująca I będzie większa od 8. Inaczej: pętla działa dopóki zmienna sterująca I jest mniejsza lub równa 8. W pętli wykonujemy dwie instrukcje: losową liczbę wpisujemy do kolejnego elementu wektora i zwiększamy zmienną sterującą I o jeden.

```
i:=1
while i≤8
LI_i:=random(6)+1
i:=i+1
LI=
(2)
(1)
(4)
(3)
(5)
(5)
(1)
(4)
```

ĆWICZENIE 11 - na dwójkowe

Utwórz funkcję, która zamieni podaną liczbę na liczbę dwójkową.

Wynik otrzymamy w postaci łańcucha tekstowego.

Zmienna N początkowo jest pusta.

W pętli WHILE wykonujemy następujące operacje:

Wyciągamy resztę z dzielenia całkowitego liczby przez 2

Dzielimy liczbę przez 2, obcinamy jej część ułamkową

Na początek napisu N doklejamy resztę zamienioną na tekst.

Poniżej kilka przykładów

```
DzDw(liczba):= N:= ""
while liczba > 0
  r:=mod(liczba; 2)
  liczba:=trunc(liczba/2)
  N:=concat(num2str(r); N)
DzDw:=N

DzDw(10)= "1010"
DzDw(102)= "1100110"
DzDw(255)= "11111111"
```

ĆWICZENIE 12 - sortowanie bąbelkowe

Napisz funkcję, która posortuje elementy w tablicy.

W tablicy T znajdują się nieuporządkowane liczby. Funkcja SORTUJ porządkuje je malejąco. Algorytm nazywamy sortowanie bąbelkowym.

Zmienna ILE zawiera liczbę elementów tablicy.

Pierwsza pętla pobiera element tablicy.

Druga pętla pobiera kolejno wszystkie następne do końca.

Jeżeli element z pierwszej pętli jest mniejszy niż element z pętli drugiej, to następuje zamiana elementów. W przeciwnym wypadku „nic nie rób”. Po zakończeniu działania drugiej pętli na pierwszym miejscu znajduje się już element największy.

Pętla pierwsza wskazuje teraz element następny, a pętla druga zaczyna wskazywać o jeden większy i cała procedura sprawdzania i zamiany się powtarza. Za każdym razem na kolejne miejsce w tablicy trafia największa znaleziona liczba.

```
T:= (2
1
6
2
3
1
3
5)

sortuj(T):= ile:=length(T)
for i∈1..ile-1
  for j∈i+1..ile
    if T_i < T_j
      x:=T_j
      T_j:=T_i
      T_i:=x
    else
      0
  sortuj:=T

sortuj(T)= (6
5
3
3
2
2
1
1)
```

ZADANIA

- Narysuj wykresy funkcji: $f(x) = \begin{cases} \sin(x) & x > 0 \\ \cos(x) & x \leq 0 \end{cases}$ $g(x) = \begin{cases} \sin(x) & 1 > x > -1 \\ x^2 - 1 & -1 < x < 1 \end{cases}$

$$h(x) = \begin{cases} 2x^3 - 4x^2 + 3x - 1 & x > 0 \\ 4x^2 - 12x + 2 & x \leq 0 \end{cases}$$

- Utwórz funkcję o nazwie POLE z dwoma parametrami: a i b . Parametry są długościami boków prostokąta, a funkcja ma liczyć powierzchnię. Na przykład $\text{POLE}(2,5) \rightarrow 10$
- Utwórz funkcję o nazwie OBWÓD z dwoma parametrami: a i b . Parametry są długościami boków prostokąta, a funkcja ma liczyć obwód. Na przykład $\text{OBWÓD}(2;5) \rightarrow 14$
- Utwórz funkcję o nazwie SINUS z jednym parametrem. W parametrze podajemy kąt (w stopniach), a funkcja zwraca wartość sinusa tego kąta. Na przykład $\text{SINUS}(90) \rightarrow 1$
- Do funkcji POLE i OBWÓD dopisz instrukcję, która będzie sprawdzać, czy nie podano liczby ujemnej. Jeśli tak, to zamieni ją automatycznie na dodatnią.
- Za pomocą pętli FOR utwórz tablicę zawierającą 10 kolejnych liczb parzystych (2 4 6 8 10 12 14 16 18 20)
- Za pomocą pętli FOR utwórz tablicę zawierającą 10 kolejnych liczb nieparzystych (1 3 5 7 9 11 13 15 17 19)
- Za pomocą pętli FOR utwórz tablicę zawierającą 10 kolejnych liczb podzielnych przez 3 (3 6 9 12 15...)
- Poprzednie trzy zadania wykonaj za pomocą pętli WHILE
- Napisz funkcję, która odwróci wszystkie litery podanego napisu (Wacek \rightarrow kecaW)
- Napisz funkcję, która zamieni nazwisko i imię w podanym napisie (Wacław Libront \rightarrow Libront Wacław)
- W tablicy dwuwymiarowej znajdują się oceny. Napisz 2 funkcję: pierwsza wyliczy średnie dla wszystkich wierszy, a druga dla wszystkich kolumn.

$$\begin{bmatrix} 3 & 3 & 1 \\ 4 & 3 & 2 \\ 5 & 5 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 2,33 \\ 3 \\ 4,67 \end{bmatrix} \quad \begin{bmatrix} 3 & 3 & 1 \\ 4 & 3 & 2 \\ 5 & 5 & 4 \end{bmatrix} \rightarrow [4 \quad 3,67 \quad 2,33]$$