

KOŁOKWIUM NR 2 – PROGRAMOWANIE PODSTAWOWE

Zadania praktyczne:

1. Zmienne

- a. Napisz program Walec v1.0 obliczający objętość walca o podanej wysokości i podanym promieniu podstawy. Zdefiniuj stałą pi o wartości 3.1415 i użyj jej w obliczeniach.
- b. Napisz program Potęga v1.0 obliczający drugą potęgę z danej liczby.li>

2. Funkcja if

- a. Napisz program Pin v1.0 sprawdzający poprawność wprowadzonego pinu i proszący o potwierdzenie (ponowne wprowadzenie pinu)
- b. Napisz program sprawdzający poprawność wprowadzonego miesiąca.

3. Pętla for

- a. Napisz program, który wypisze na ekran liczby podzielne przez 3 i podzielne przez 5.
- b. Napisz program, który sprawdza, czy podana liczba jest liczbą pierwszą
- c. Napisz program, który z podanego przedziału (od jeden) wypisuje liczby pierwsze.

4. Pozostałe funkcje:

- a. Napisać rekurencyjną funkcję

```
int silnia (int n);
```

do liczenia funkcji silnia

$n!$

według wzoru:

```
silnia(0) = 1 silnia(n) = n * silnia(n-1)
```

b.*

Znaleźć rozwiązanie problemu Wież z Hanoi. Ta starożytna łamigłówka pochodzi z pewnego klasztoru w Tybecie:

Przypuśćmy, że mamy trzy wieże lub, skromniej, trzy kołki, A, B i C. Na pierwszym kołku, A, znajdują się trzy krążki nanizane w porządku malejących wielkości, podczas gdy pozostałe kołki są puste. Chcemy przenieść krążki z kołka A na B, być może używając do tego kołka C. Według reguł gry krążki można przenosić po jednym na raz i w żadnej chwili krążek większy nie może być umieszczony na wierzchu mniejszego.

Wskazówka: Przedstawiony tu algorytm radzi sobie z przeniesieniem N krążków z kołka A przez C na kołek B następująco. Najpierw sprawdza, czy $N=1$, w którym to przypadku po prostu przenosi na miejsce przeznaczenia ten jedyny krążek, z którym miał się uporać

(lub, dokładniej, podaje opis jednego ruchu, który to robi) i natychmiast powraca. Jeśli $N > 1$, algorytm najpierw przenosi górne $N-1$ krążków z kołka A na kołek "pomocniczy" C, używając rekurencyjnie *tego samego* podprogramu; potem bierze jedyny krążek pozostawiony na kołku A (musi to być krążek największy - dlaczego?) i przenosi go na ostateczne miejsce przeznaczenia, na kołek B; potem, znowu rekurencyjnie, przenosi $N-1$ krążków, które poprzednio "przechował" na kołku C, na ich ostateczne miejsce, na kołek B. Napisz rekurencyjną funkcję

```
void hanoi (int N, char A, char B, char C);
```

realizującą ten algorytm i wydrukować wynik dla trzech krążków.

5. Tablice

- Napisz program, który pobierze żadaną ilość pomiarów, wypisze je a następnie zapyta czy wprowadzone pomiary są poprawne. Jeżeli nie, wczytanie pomiarów rozpocznie się od nowa.
- Napisz program, który pobierze do tablicy żadaną ilość liczb i wypisze liczby większe od średniej z wszystkich liczb.
- Napisz program, który pobierze od użytkownika jego imię i powita np.: Witaj, Jan.
- Napisz program, który pobierze hasło i znajdzie w nim cyfry.
- Napisz program, który pobierze hasło i znajdzie w nim duże litery. Skorzystaj z numerów znaków z tablicy ASCII, aby rozpoznać duże litery.

6. Wskaźniki

- Napisz program, w którym wczytasz i wypiszesz na ekran elementy tablicy przy użyciu wskaźników oraz obliczysz ich średnią.
- Napisz program, w którym wczytasz elementy tablicy, obliczysz ich średnią oraz wpiszesz elementy większe od średniej. Wszystko przy pomocy wskaźników.

ZADANIE TEORETYCZNE:

Instrukcja grupująca to inaczej:

instrukcja przypisania

instrukcja blokowa

instrukcja warunkowa

Blok instrukcji jest ciągiem instrukcji ujętych w:

" " (cudzysłów)

{ } (nawiasy klamrowe)

[] (nawiasy prostokątne)

Polecenie IF opisuje:

instrukcję wyboru

instrukcję warunkową

instrukcję blokową

Instrukcje warunkowe można:

zagnieżdżać

stopniować

obie odpowiedzi są prawidłowe

Instrukcja switch:

jest instrukcją, która wybiera sekwencje instrukcji

jest instrukcją, która pozwala dokonać wyboru jednej z wielu sekwencji instrukcji

jest instrukcją, która pozwala dokonać wyboru jednej z wielu zmiennych

Instrukcja pętli, w której blok instrukcji realizowany jest cyklicznie, w przypadku gdy wartość logiczna wyrażenia sterującego pętlą wynosi true (jest różna od zera) to:

while

while to

for

Pętla do...while jest podobna do pętli while z tą różnicą:

że warunek sprawdzany w trakcie wykonywania pętli

że warunek sprawdzany jest przed każdym wykonaniem pętli

że warunek sprawdzany jest po każdym wykonaniu pętli

Pętla for jest rozwinięciem

instrukcji wyboru

instrukcji blokowej

pętli while

Instrukcję for można przedstawić posługując się instrukcją while w następujący sposób:
gdzie: W1 ? ustawienie początkowych wartości (inicjalizacja) zmiennej lub zmiennych sterujących pętlą; W2 - warunek pozostania w pętli; W3 - instrukcja lub instrukcje modyfikacji zmiennych sterujących; I ? blok instrukcji realizowany w ramach pętli

W1 while (W2) {I W3}

W1 while {W2, I W3}

W1 while [W2] [W3]

Instrukcja continue powoduje:

w pętli for zakończenie realizacji bieżącej iteracji

w pętlach while i do while pominięcie dalszych instrukcji w bloku

obie odpowiedzi są prawidłowe