

## Ćwiczenia – lista zadań nr 2

### Zadanie 1

- a) Dla przedstawionego poniżej fragmentu programu narysuj schemat blokowy. Przeanalizuj działanie tego programu i odtwórz dokładnie napisy, które zostaną wypisane na ekranie po uruchomieniu tego programu.

```
int wiersz, kolumna;

    wiersz = 1;
    kolumna = 1;
wroc:
    if (wiersz>=10) goto dalej;

    printf("%3d", kolumna);
    if (wiersz == kolumna) {
        wiersz++;
        kolumna=0;
        printf("\n");
    }
    kolumna++;
    goto wroc;
dalej:
```

- b) W programie z poprzedniego punktu wyeliminuj instrukcje `goto etykieta` poprzez użycie pętli `while`.  
Wskazówka: W pętli `while` użyj warunku, który jest zaprzeczeniem warunku z pierwszej instrukcji `if`.
- c) W programie z punktu b) zastąp pętlę `while` pętlą `for`.
- d) Napisz program drukujący dokładnie takie same napisy jak programy z poprzednich punktów, w którym zostaną użyte dwie zagnieżdżone pętle `for`. W pierwszej zmiennej sterującą powinna być zmienna `wiersz`, a w drugiej zmiennej sterującą powinna być zmienna `kolumna`.
- e) W programie z punktu b) zastąp pętlę `while` za pomocą pętli `do { } while`. Który z uzyskanych w punktach od a) do e) wariantów programu uważasz za najłatwiejszy do zrozumienia. Uzasadnij odpowiedź.

### Zadanie 2

- a) Napisz program, który wczytuje z klawiatury wysokość trójkątów, a następnie rysuje na ekranie trójkąty złożone ze znaków `#` podobnie jak na przedstawionym poniżej rysunku. Wskazówka: Żeby uzyskać pożądany kształt trójkąta na ekranie, w każdym wierszu przed znakami `#` trzeba wydrukować odpowiednią liczbę spacji, które na rysunku są niewidoczne.

```

#           #####           #
##          #####          ###
###         ###           #####
####        ##           #####
#####       #           #####
```

- b) Napisz program, który wczytuje z klawiatury wysokość liter, a następnie rysuje na ekranie za pomocą znaków # duże litery M N Z o podanej wysokości. (zob. przykład liter dla wysokości 5 znaków).

```

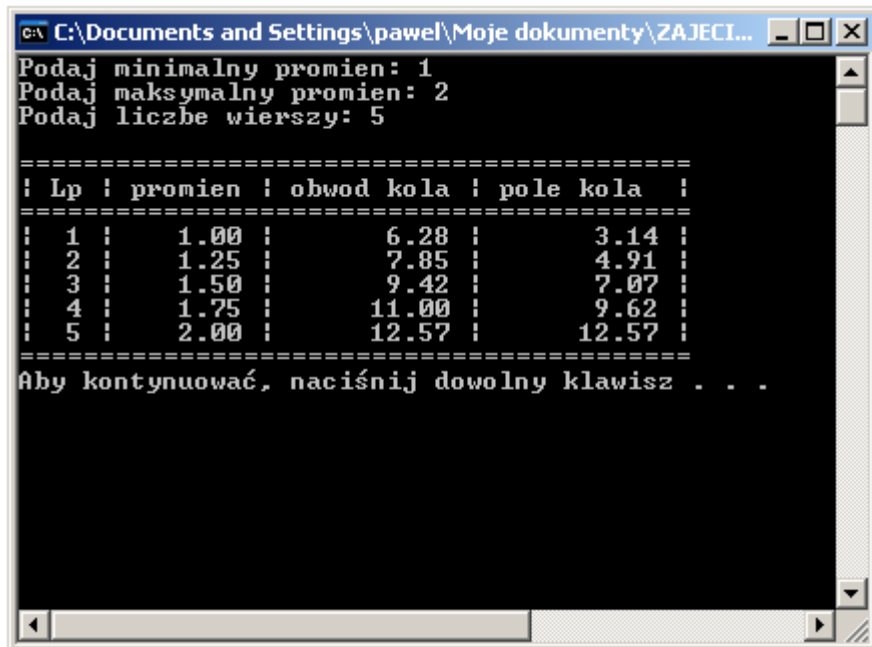
#         #         #         #         #####
##        ##        ##        #
# #       # #       # #       #
#  #  #  #  #       #  ##      #
#         #         #         #####

```

### Zadanie 3

- a) Napisz funkcję, która wczytuje z klawiatury minimalny i maksymalny promień koła oraz liczbę wierszy tabeli, w której będą wypisane promienie i odpowiadające im obwód oraz pole koła. Tabela z ramkami oraz nagłówkiem ma zawierać cztery kolumny: numer wiersza, promień koła, obwód koła, pole koła. Wzór tabelki dla 5 wierszy pokazano na rysunku poniżej.

Do drukowania tabelki użyj funkcji `printf` z biblioteki `stdio.h`.



- b) Napisz drugi wariant tej funkcji, w którym drukowanie tabelki będzie zrobione za pomocą wysyłania znaków do strumienia `cout`. (należy wykorzystać bibliotekę `iostream`).

### Zadanie 4

Proszę napisać funkcję, która wczytuje z klawiatury pożądaną dokładność obliczeń  $\epsilon$  ( $\epsilon < 0.1$ ), a następnie oblicza zadaną dokładnością wartość sumy następujących szeregów:

$$S1 = 4 * \sum_{k=1}^{\infty} \left[ (-1)^{k+1} * \frac{1}{2k-1} \right] = 4 * \left[ \frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots \right]$$

$$S2 = \sum_{k=0}^{\infty} \frac{1}{k!} = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots = \frac{1}{1} + \frac{1}{1} + \frac{1}{1*2} + \frac{1}{1*2*3} + \dots$$

**Wskazówki:**

- Sumowanie szeregu należy zakończyć, gdy wartość bezwzględna pojedynczego wyrazu szeregu jest mniejsza niż zadana dokładność obliczeń *eps*.
- Suma szeregu *S1* jest zbieżna do wartości stałej  $\pi$ , suma szeregu *s2* jest zbieżna do wartości stałej *e*.

**Zadanie 5**

Przeanalizuj przykładowe programy zawarte w poniższej tabeli. Dla każdego programu zaznacz zakres widoczności wszystkich zmiennych, a następnie wpisz w prawej kolumnie napisy, które zostaną wypisane na ekranie po uruchomieniu programu.

**Uwaga:**

Szczególną uwagę proszę zwrócić na kwalifikator globalności

::identyfikator\_globalny,

który może być stosowany wyłącznie w programach pisanych w języku C++.

Program w C/C++:	Wynik działania programu (wydruk na ekranie)
<b>Przykład 1:</b> <pre>#include &lt;stdio.h&gt;  int A = 10;  void Funkcja() {     int A = 15;     printf("A1 = %d\n", A); }  int main() {     printf("A2 = %d\n", A);     {         int A = 7;         Funkcja();         printf("A3 = %d\n", A);     }     printf("A4 = %d\n", A);     return 0; }</pre>	<pre>A2 = 10 A1 = 15 A3 = 7 A4 = 10</pre>

**Przykład 2:**

```
#include <stdio.h>

int R = 5;

int main()
{
    int R = 30;
    {
        int R = 100;
        printf("R1 = %d\n", R);
    }
    printf("R2 = %d\n", R);
    return 0;
}
```

**Przykład 3:**

```
#include <stdio.h>

int K = 3;

void funkcja()
{
    printf("K1 = %d\n", K);
}

int main()
{
    int K = 20;
    printf("K2 = %d\n", K);
    funkcja();
    return 0;
}
```

**Przykład 4:**

```
#include <stdio.h>

int K = 3;

int fun()
{
    return (K);
}

int main()
{
    int K = 20;
    printf("K = %d\n", K + fun());
    return 0;
}
```

**Przykład 5:**

```
#include <stdio.h>

int R = 5;

int main()
{
    int R = 30;
    {
        int R = 100;
        printf("R1 = %d\n", R + ::R);
    }
    printf("R2 = %d\n", ::R+R);
    return 0;
}
```