

Zajęcia 3

Temat: Pętla for, while, do-while 1

Czas trwania: 2x45 min

Cel zajęć:

projektuje i programuje proste problemy z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, testuje poprawność programów dla różnych danych, posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

Efekty:

- zna różne systemy liczbowe i potrafi dokonać zamiany,
- umie napisać program z wykorzystaniem instrukcji warunkowej i iteracyjnej,

Formy i metody pracy: praca samodzielna, omówienie, wykład

Zadania do wykonania na zajęciach	Treści programowe
1. Dwójki	M.1, P.2.10
2. Litery w liczbie	M.2, P.2.10, A.1, A.3.1
3. Małpki na wybiegu	P.2.10
4. Szachownica	M.1, P.2.10
5. Najbliższy palindrom	P.2.10, A.1, A.3.1

Materiały do zajęć:

<https://www.main2.edu.pl/main2/courses/show/6/14/>

<https://www.main2.edu.pl/main2/courses/show/6/17/> (część dot. pętli)

Zadania do wykonania w domu:

Zadania z ogólnodostępnych serwisów z zadaniami.

Maksymalna różnica:

<https://szkopul.edu.pl/problemset/problem/ewrYvFHd7H87PgwhxdWtmaBv/site/?key=statement>

Prostokąt:

<https://szkopul.edu.pl/problemset/problem/ihvm-DVlj3xdP2fKuCm4cJeO/site/?key=statement>

Ciąg sum częściowych:

<https://szkopul.edu.pl/problemset/problem/B6X2pKq-yacHF3AvjcDs48Lk/site/?key=statement>

ZADANIA I ROZWIĄZANIA

Zadanie 1. Dwójki

Limit pamięci: 64MB

Dana jest liczba naturalna x ($x < 10^{18}$). Dla każdego x wypisz liczbę potęg 2 mniejszych bądź równych każdego z x .

Przykład

Wejście 10 Wyjście 4	Wejście 64 Wyjście 7
-------------------------------	-------------------------------

Rozwiązanie

Zadanie wymaga użycia pętli `while`.

W zadaniu zaczynamy od wartości potęgi równej (dla x równego 0). Tak długo mnożymy kolejne dwójki, aż uzyskany wynik będzie większy niż zadana liczba. Jednocześnie zliczamy wykonane mnożenia.

```
potęga ← 1, liczba_potęg ← 0
wczytaj x
dopóki potęga < x wykonaj
    potęga ← potęga • 2
    liczba_potęg ← liczba_potęg + 1
wypisz liczba_potęg - 1
```

Zadanie 2. Litery w liczbie

Janek bada właściwości liczb i ostatnio szuka takich liczb, które zapisane w postaci szesnastkowej zawierają co najmniej jedną literę.

Wejście

Pierwszy wiersz danych zawiera liczbę naturalną mniejszej od trylionu.

Wyjście

Wypisz komunikat TAK, jeśli w postaci szesnastkowej występują litery lub NIE – jeśli ich brak.

Wejście 54 Wyjście NIE	Wejście 255 Wyjście TAK
---------------------------------	----------------------------------

Rozwiązanie

Zadanie wymaga użycia pętli `while`.

Wykorzystamy fakt, że aby zamienić liczbę z systemu dziesiętnego na liczbę w systemie o dowolnej innej podstawie wystarczy dzielić przez nową podstawę. Jednak zamiast

wypisywać reszt, wystarczy sprawdzić, czy w wyniku wystąpiłaby litera (reszta wynosi co najmniej 10).

```
wartownik ← 0
wczytaj x
dopóki x > 0 i wartownik = 0 wykonaj
    jeżeli x mod 16 ≥ 10 to
        wartownik ← 1
    x ← x div 16
jeżeli wartownik = 1 to
    wypisz „TAK”
w przeciwnym wypadku
    wypisz „NIE”
```

Zadanie 3. Małpy na wybiegu

Dostępna pamięć: 32MB

Mały Bitek trenuje małpki. Umie już je ustawić w jednym rzędzie. Planuje teraz kolejne sztuczki. Na noc jednak Bitek musi schować małpki na wybiegu. Pomóż mu!

Wejście

W pierwszym i jedynym wierszu wejścia znajduje się jedna liczba całkowita n ($1 \leq n \leq 10^3$) – liczba małpek.

Wyjście

Wypisz n znaków @ obok siebie otoczonych płotkiem ze znaków #.

Przykład

Wejście	Wyjście
5	##### #@#@#@#@# #####

Rozwiązanie

Zadanie pozwala na trzykrotne użycie pętli `for`.

Zauważmy, że będziemy wypisywać kolejne znaki @ oraz #. W pierwszej linii musimy wypisać $n+2$ kratki (ogrodzenie jest większe z lewej i prawej strony o jeden element). W drugiej linii wypiszemy jedną kratkę, n małpek i znowu jedną kratkę. W trzeciej linii ponownie musimy wypisać $n+2$ kratki.

```
wczytaj n
dla i=1 do n+2 wykonaj
    wypisz #
przejdź do nowej linii
wypisz #
dla i=1 do n wykonaj
    wypisz @
wypisz #
przejdź do nowej linii
dla i=1 do n+2 wykonaj
```

wypisz #

Zadanie 4. Szachownica

Dostępna pamięć: 32MB

Napisz program, który dla podanej na standardowym wejściu liczby całkowitej n , narysuje szachownicę z cyfr 0 i 1 o boku n .

Wejście

Jedyny wiersz danych zawiera liczbę całkowitą n ($1 \leq n \leq 200$).

Wyjście

Program powinien wypisać szachownicę o wielkości n .

Przykład

Wejście	Wyjście
5	01010 10101 01010 10101 01010

Rozwiązanie

Zadanie wymaga użycia zagnieżdżonej pętli, na przykład `for`.

Zewnętrzna pętla będzie odpowiadać za rysowanie kolejnych wierszy (iteracja y), zaś wewnętrzna kolejnych elementów w wierszu (iteracja x). Zauważmy, że cyfry 0 występują zawsze, gdy numer wiersza i kolumny ma tę samą parzystość. W zadaniu wykorzystamy fakt, że suma dwóch liczb o tej samej parzystości jest parzysta.

```
wczytaj n
dla y=1 do n wykonaj
    dla x=1 do n wykonaj
        jeżeli (x + y) mod 2 = 0
            wypisz 0
        w przeciwnym wypadku
            wypisz 1
    przejdź do nowej linii
```

Analizując warunek możemy dokonać uproszczenia:

```
wczytaj n
dla y=1 do n wykonaj
    dla x=1 do n wykonaj
        wypisz (x + y) mod 2
    przejdź do nowej linii
```

Zadanie 5. Najbliższy palindrom

Mały Bitek przeczytał na Wikipedii: Palindrom ([gr. palindromeo](#) – biec z powrotem) – wyrażenie brzmiące tak samo czytane od lewej do prawej i od prawej do lewej. Janek szybko

ułożył sobie jeden palindrom (biorąc pod uwagę tylko litery): „U Izydy żądze na wyrku co noc ukrywane, zdąży Dyziu?”. Potem zaczął wymyślać całą masę następnych. Niestety – dużo gorzej mu idzie z liczbami. Chciałby szybko zamienić dowolną liczbę (jeśli nie jest ona palindromem) na najbliższy większy od niej palindrom. Czy mu pomożesz?

Wejście

Pierwszy wiersz danych zawiera liczbę całkowitą k ($10 \leq k \leq 1\,000\,000$) do zamiany na palindrom.

Wyjście

Program powinien wypisać w jedną liczbę całkowitą – najmniejszą możliwą liczbę nieujemną, którą należy dodać do danej liczby, aby otrzymać palindrom.

Przykład

Wejście 150 Wyjście 1	Wejście 55555 Wyjście 0	Wejście 142 Wyjście 9
--------------------------------	----------------------------------	--------------------------------

Rozwiązanie

Zacznijmy od sprawdzenia, czy podana liczba k jest palindromem. W tym celu wyznaczmy liczbę odwrócona_k (z odwróconą kolejnością cyfr w liczbie). Będziemy „odrywać” po jednej cyfrze z liczby o wartości k (operacje mod 10 oraz div 10).

```
k' ← k
odwrócona_k ← 0
dopóki k' > 0 wykonaj
    odwrócona_k ← odwrócona_k • 10 + k' mod 10
    k' ← k' div 10
```

Zauważmy, że w przypadku, gdy odwrócona_k jest równa k mamy do czynienia z palindromem. Możemy więc dodawać kolejne wartości tak długo, aż otrzymamy palindrom.

```
wartownik ← 0
i ← 0
dopóki wartownik = 0 wykonaj
    OBLICZ odwrócona_k dla wartości k+i
    jeżeli odwrócona_k = k+i
        wartownik ← 1
    w przeciwnym wypadku
        i ← i + 1
wypisz i
```