

# Zadanie 1. Wyznaczanie daty Wielkanocy

Wielkanoc jest tak zwanym świętem ruchomym - oznacza to, że każdego roku występuje w inny dzień. Definicja mówi, iż przypada ona w pierwszą niedzielę po pierwszej pełni Księżyca przypadającej po równonocy wiosennej. W 1991 roku Jean Meeus opublikował sposób na obliczanie daty Wielkanocy.

Aby obliczyć datę Wielkanocy:

- Tworzymy zmienne  $a, b, c, d, e, f, g, h, i, k, l, m, p$ , zmienną przechowującą podany przez użytkownika rok, a także obliczony dzień i miesiąc
- Pobieramy rok od użytkownika
- Za  $a$  podstaw resztę z dzielenia roku przez 19
- Za  $b$  podstaw zaokrąglony w dół wynik dzielenia roku przez 100
- Za  $c$  podstaw resztę z dzielenia roku przez 100
- Za  $d$  podstaw zaokrąglony w dół wynik dzielenia  $b$  przez 4
- Za  $e$  podstaw resztę z dzielenia  $b$  przez 4
- Za  $f$  podstaw zaokrąglony w dół wynik działania  $((b + 8) / 25)$
- Za  $g$  podstaw zaokrąglony w dół wynik działania  $((b - f + 1) / 3)$
- Za  $h$  podstaw resztę z dzielenia przez 30 wyrażenia  $(19 * a + b - d - g + 15)$
- Za  $i$  podstaw zaokrąglony w dół wynik dzielenia  $c$  przez 4
- Za  $k$  podstaw resztę z dzielenia  $c$  przez 4
- Za  $l$  podstaw resztę z dzielenia przez 7 wyrażenia  $(32 + 2 * e + 2 * i - h - k)$
- Za  $m$  podstaw zaokrąglony w dół wynik dzielenia przez 451 wyrażenia  $(a + 11 * h + 22 * l)$
- Za  $p$  podstaw resztę z dzielenia przez 31 wyrażenia  $(h + l - 7 * m + 114)$

Aby wyznaczyć dzień Wielkanocy do  $p$  należy dodać 1, z kolei do wyznaczania miesiąca trzeba zaokrąglając w dół podzielić  $(h + l - 7 * m + 114)$  przez 31. Jeżeli miesiąc wyszedł nam 4, wówczas Wielkanoc odbędzie się w kwietniu. W każdym innym przypadku w marcu.

- a) Przystąp do implementacji algorytmu. Przed rozpoczęciem pracy spróbuj rozpisać każde działanie na kartce. Na wyjściu zwróć datę w formacie DD miesiąc RRRR np. dla roku 2015 Twój program powinien zwrócić 5 kwietnia 2015.
- b) Spróbuj rozrysować schemat blokowy dla takiego algorytmu.

## Zadanie 2. Biedny Camel Case

CamelCase jest konwencją notacji przyjętą w wielu językach programowania, takich jak Java i C#, gdzie służy do zapisu nazw zmiennych, procedur, klas, itp. (w Pythonie korzysta się z tej notacji np. w nazewnictwie klas). Nazwa notacji camelCase wywodzi się z faktu, że przy jej zastosowaniu wielkie litery w połączonych ze sobą słowach przypominają kształtem garby wielbłąda (ang. camel). Konwencja ta została najprawdopodobniej użyta po raz pierwszy w języku programowania Smalltalk.

W latach 90-00 modne było tworzenie opisów w GG przy pomocy notacji przypominającej CamelCase – jedna litera duża, druga mała np. JaKsłęCiEsZę (Jak się cieszę).

Zaimplementuj algorytm, który dla wprowadzonego przez użytkownika ciągu znaków zwróci mu biednego CamelCase'a zgodnie z przykładem w tekście. Załóż, że program nie będzie wczytywał polskich znaków, a spacja nie jest liczona jako litera (jest przepisywana).

Jak się cieszę -> JaK słę CiEsZę

# Zadanie 3. Spacje CamelCase

*Zaimplementuj poniższy algorytm.*

W celu zaoszczędzenia ilości znaków w krótkich wiadomościach tekstowych (SMS) nie pisze się spacji, a każdy wyraz rozpoczyna się wielką literą. Twoim zadaniem jest otrzymany tekst przerobić zgodnie z powyższym trendem.

## Wejście

Na wejściu znajduje się dowolny tekst bez polskich znaków.

## Wyjście

Tekst wprowadzony z wejścia, ale bez spacji. Ponadto każdy wyraz poprzedzony na wejściu spacją zaczyna się wielką literą.

## Przykład

### Input:

Dzisiaj jest czwartek,  
A jutro będzie piątek.

### Output:

DzisiajJestCzwartek,  
AJutroBedziePiatek.

# Zadanie 4. Spacje CamelCase odwrotność

*Zaimplementuj poniższy algorytm.*

W celu zaoszczędzenia ilości znaków w krótkich wiadomościach tekstowych (SMS) nie pisze się spacji, a każdy wyraz rozpoczyna się wielką literą. Twoim zadaniem jest odkodować otrzymany tekst i przerobić tak, aby z wersji bez spacji otrzymać normalny tekst (wyrazy nie rozpoczynające zdanie mają zostać zapisane z małej litery). Załóż, że pojedyncze zdanie znajduje się w pojedynczej linii.

## Wejście

Na wejściu znajduje się dowolny tekst bez polskich znaków pozbawiony spacji.

## Wyjście

Tekst wprowadzony z wejścia ze spacjami.

## Przykład

**Input:**

DzisiajJestCzwartek,

**Output:**

Dzisiaj jest czwartek,

Spróbuj napisać klasę CamelCase, która będzie posiadać metody pozwalające na dokonanie operacji na napisach jak w zadaniach 2-4.

# Zadanie 5. Dzień tygodnia

*Zaimplementuj algorytm wyznaczający dzień tygodnia za  $n$  dni.*

Dni tygodnia będziemy oznaczać skrótami: Pn, Wt, Sr, Czw, Pt, Sb, Nd.

## Wejście

Przypadek składa się ze skrótu dnia tygodnia i liczby  $n$  ( $1 \leq n \leq 10^9$ ).

## Wyjście

Skrót dnia tygodnia, który nastąpi za  $n$  dni licząc od dnia tygodnia podanego na wejściu.

## Przykład

**Input:**

Czw 8

**Output:**

Pt

# Zadanie 6. Imieniny Jasia

*Przeczytaj poniższy tekst i zaimplementuj algorytm pozwalający Jasiowi sprawdzić czy zostaną mu jakieś cukierki, a jeżeli tak to ile.*

Jaś ma pojutrze imieniny. Jak dla każdego małego chłopca, jest to bardzo miły dzień w jego życiu. Na pewno dostanie mnóstwo prezentów, słodkości i innych pyszności. Jednak Jaś chodzi do szkoły, a jego imieniny wypadają akurat w dzień, który żadnym świętem nie jest i musi do niej pójść. Nie to jednak jest w tym najgorsze, że trzeba tam iść, zamiast cały dzień świętować objadając się pysznościami przygotowanymi przez mamę. Najgorsze jest to, że jak każdy inny mały chłopiec w szkole, Jaś musi rozdać swoim kolegom i koleżankom cukierki w czasie jednej z lekcji. Jest to zadanie, którego Jaś szalenie wręcz nie lubi. Jednak nie może nic z tym zrobić – taka jest tradycja, a nie chce okłamywać mamy, że rozdał cukierki, naprawdę ich nie rozdając. Zresztą... wstyd przed całą klasą – mieć imieniny i nie mieć cukierków – nie, nie, absolutnie nie wchodzi to w grę.

Mama już zakupiła odpowiednią ilość cukierków. Ponieważ ich rozdawanie jest ogromnie ważne, mama już dziś położyła cukierki tuż przy plecaku Jasia w jego pokoju, aby na pewno ich nie zapomniał. A tymczasem on nie może przez to spać. Cukierki, przypominając zapachem o swojej obecności, przywołują u Jasia strasznie niemiłe wspomnienia zeszłorocznych imienin i cukierków w klasie, kiedy na oczach całej podśmiewającej się klasy musiał się całować z panią nauczycielką...

Jaś postanowił odpędzić złe myśli zabijając w jakiś sposób czas. Wziął torebkę z cukierkami i postanowił je policzyć. Policzył jednak bardzo szybko i znów złe myśli przysły mu do głowy. Zdecydował więc, że musi się dowiedzieć, czy w tym roku też zostaną dla niego cukierki, jak w zeszłym, czy może tym razem mama się poprawiła w obliczeniach. Jaś wiedział, ile w jego klasie jest osób i wiedział, że nie może nikogo wyróżnić – każdy musi dostać taką samą liczbę cukierków, oprócz niego samego, który nie je ich wtedy w szkole. Był dobrym kolegą, więc każdemu chciał dać jak najwięcej. Jednak, jeśli nie dało się rozdać w taki sposób wszystkich cukierków, te które zostały Jaś brał z powrotem do domu i mógł sam je zjeść.

Obliczenie, czy coś mu zostanie było ponad jego siły, biednego, małego chłopca. Poprosił więc Ciebie o pomoc. Wiedząc ile osób liczy klasa Jasia oraz ile cukierków kupiła mama oblicz, czy po ich rozdaniu wśród kolegów i koleżanek Jasia, zostanie jeszcze coś dla niego na wieczór, a jeżeli tak to ile.

## Wejście

Wejście składa się z dwóch liczb całkowitych  $L$  i  $C$  ( $1 \leq L, C \leq 109$ ) oznaczających odpowiednio liczbę osób w klasie Jasia oraz liczbę cukierków, które kupiła mama.

## Wyjście

Należy wypisać w osobnej linii słowo TAK oraz liczbę cukierków, jeśli coś dla Jasia zostanie lub NIE, jeśli rozda on dokładnie wszystkie cukierki.

## Przykład

3 2 => NIE

55 22 => TAK 13

# Zadanie 7. Weryfikacja dowodu osobistego

*Przeczytaj poniższy tekst, następnie zaimplementuj podany algorytm, który po wczytaniu od użytkownika jego numeru dowodu zwaliduje go. Spróbuj narysować schemat blokowy dla algorytmu.*

Obowiązujący (...) wzór dowodu osobistego został określony rozporządzeniem Rady Ministrów (...). Obecny wzór określa rozporządzenie Ministra Spraw Wewnętrznych z dnia 29 stycznia 2015 r. w sprawie wzoru dowodu osobistego oraz sposobu i trybu postępowania w sprawach wydawania dowodów osobistych, ich utraty, uszkodzenia, unieważnienia i zwrotu. Dokument ma postać wielowarstwowej karty poliwęglanowej o wymiarach 85,60 × 53,98 mm (...), zawierającej podstawowe informacje o jej posiadaczu, dacie wydania, organie wydającym i terminie ważności. Informacje zapisane są zarówno w sposób jawny, jak i zakodowany. Dowód osobisty jest jednocześnie dokumentem uprawniającym obywateli polskich do przekraczania granic państw członkowskich Unii Europejskiej, państw Europejskiego Obszaru Gospodarczego nienależących do Unii Europejskiej oraz państw niebędących stronami umowy o Europejskim Obszarze Gospodarczym, których obywatele mogą korzystać ze swobody przepływu osób na podstawie umów zawartych przez te państwa ze Wspólnotą Europejską i jej państwami członkowskimi, uznających ten dokument za wystarczający do przekraczania ich granicy.

Od listopada 2013 roku został wprowadzony do obiegu zmodyfikowany dowód osobisty. Dokument powstał na bazie poprzedniego wzoru dowodu. Modyfikacja nie spowodowała zmiany wzoru dokumentu, jednak wprowadzono kilka zmian w zabezpieczeniach. Nowe dokumenty rozpoczynają się serią AXI od numeru x00001.

Numer „plastikowego” dowodu osobistego składa się z trzech liter i sześciu cyfr, z których jedna (pierwsza cyfra) to cyfra kontrolna.

Aby sprawdzić poprawność numeru i serii dowodu, należy najpierw zamienić litery na liczby, przypisując literom odpowiednio:

A	B	C	D	E	F	G	H	I	J	K	L	M	N
10	11	12	13	14	15	16	17	18	19	20	21	22	23
O	P	Q	R	S	T	U	V	W	X	Y	Z		
24	25	26	27	28	29	30	31	32	33	34	35		

Kolejnym krokiem jest wymnożenie tak uzyskanego ciągu liczb przez określone wagi. Wagi kolejnych liczb to: 7, 3, 1, 9, 7, 3, 1, 7, 3. Reszta z dzielenia sumy iloczynów przez 10 powinna być równa 0 (zero).

# Zadanie 8. Weryfikacja numeru PESEL

*Na podstawie poniższego tekstu zaimplementuj algorytm weryfikujący numer PESEL.*

Jan Kowalski musi wpisać do systemu szpitalnego dane osobowe pacjenta, oprócz imienia i nazwiska musi również wpisać PESEL pacjenta. Jakież było jego zdziwienie, gdy spostrzegł, że pewnych pacjentów system nie przyjmował z powodu wadliwego PESELu.

Aby sprawdzić czy dany PESEL jest prawidłowy należy wykonać następujące działania:

- Pierwszą cyfrę mnożymy przez 1,
- drugą cyfrę mnożymy przez 3,
- trzecią cyfrę mnożymy przez 7,
- czwartą cyfrę mnożymy przez 9,
- piątą cyfrę mnożymy przez 1,
- szóstą cyfrę mnożymy przez 3,
- siódmą cyfrę mnożymy przez 7,
- ósmą cyfrę mnożymy przez 9,
- dziewiątą cyfrę mnożymy przez 1,
- dziesiątą cyfrę mnożymy przez 3,
- jedenastą cyfrę mnożymy przez 1.

Tak uzyskane 11 iloczynów dodajemy do siebie. Jeśli ostatnia cyfra tej sumy jest zerem to podany PESEL jest prawidłowy. Przykład dla numeru PESEL 44051401458

$$\begin{aligned} &4*1 + 4*3 + 0*7 + 5*9 + 1*1 + 4*3 + 0*7 + 1*9 + 4*1 + 5*3 + 8*1 = \\ &= 4 + 12 + 0 + 45 + 1 + 12 + 0 + 9 + 4 + 15 + 8 = 110. \end{aligned}$$



# Zadanie 9. Odwrotna notacja polska

Przeczytaj poniższy opis algorytmu, następnie spróbuj go zaimplementować. Zadanie spróbuj rozwiązać po dokładnym omówieniu stosów.

**Odwrotną notację polską ONP** (ang. RPN – Reverse Polish Notation), zwana często również notacją Postfix, wymyślono w celu zapisywania dowolnych wyrażeń arytmetycznych bez nawiasów. W normalnym zapisie arytmetycznym operatory znajdują się pomiędzy argumentami:

$$2 + 2 \quad 6 - 4 \quad 3 * 5 \quad 12 / 3$$

Operatory posiadają priorytety, czyli "ważność" oraz łączność w prawo lub w lewo. Jeśli w wyrażeniu wystąpią operatory o różnych priorytetach, to najpierw zostaną wykonane te ważniejsze:

$$3 + 5 * 2 = 3 + 10 = 13$$

Jeśli chcemy zmienić kolejność wykonywania działań, musimy używać nawiasów:

$$(3 + 5) * 2 = 8 * 2 = 16$$

W ONP problem ten nie występuje. Operator zawsze występuje po swoich argumentach:

$$2 \ 2 + \quad 6 \ 4 - \quad 3 \ 5 * \quad 12 \ 3 /$$

Dzięki tej prostej zasadzie nawiasy stają się zbędne:

$$3 + 5 * 2 \rightarrow 3 \ 5 \ 2 * + = 3 \ 10 + = 13$$

$$(3 + 5) * 2 \rightarrow 3 \ 5 + 2 * = 8 \ 2 * = 16$$

Do obliczenia wartości wyrażenia zapisanego w ONP potrzebujemy stosu. Zasada jest następująca:

Wyrażenie ONP przeglądamy od strony lewej do prawej. Jeśli napotkamy liczbę, to umieszczamy ją na stosie. Jeśli napotkamy operator, to ze stosu pobieramy dwie ostatnie liczby, wykonujemy na nich działanie zgodne z napotkanym operatorem i wynik umieszczamy z powrotem na stosie. Gdy wyrażenie zostanie przeglądnięte do końca, na szczycie stosu będzie znajdował się jego wynik.

Wyrażenie ONP	Element	Operacja	Stos
3 5 2 * +			---
<b>3</b> 5 2 * +	<b>3</b>	na stos	3 ---
3 <b>5</b> 2 * +	<b>5</b>	na stos	5 3 ---
3 5 <b>2</b> * +	<b>2</b>	na stos	2 5

			3 ---
3 5 2 * +	*	pobierz 2 i 5 mnóż 5 * 2 wynik na stos	10 3 ---
3 5 2 * +	+	pobierz 10 i 3 dodaj 3 + 10 wynik na stos	13 ---

Notacja ONP jest szeroko wykorzystywana w kompilatorach języków wysokiego poziomu. Istnieją również języki, które do obliczeń stosują jedynie ONP – np. Forth.

Przed przystąpieniem do zaprojektowania algorytmu ONP musimy poczynić pewne ustalenia. Dla prostoty umawiamy się, że używać będziemy tylko czterech operatorów arytmetycznych:

- + – dodawanie
- - – odejmowanie
- \* – mnożenie
- / – dzielenie

Wyrażenie musi być poprawne – algorytm nie sprawdza jego poprawności.

Każdy element będzie wprowadzany w osobnym wierszu – w ten sposób pozbędziemy się problemu analizowania tekstu pod kątem zawartości w nim liczb i operatorów. W rzeczywistości wyrażenie zawarte w wierszu zostałoby najpierw rozbite na elementy składowe – liczby i operatory – a następnie elementy te zostałyby użyte do obliczenia wartości wyrażenia wg naszego algorytmu.

Liczby muszą mieć postać akceptowaną przez dany język programowania.

Ostatnim elementem wyrażenia jest znak "=". Powoduje on zakończenie obliczeń i wyprowadzenie wyniku ze stosu.

W algorytmie będziemy musieli rozpoznawać, czy wprowadzony element jest liczbą, czy też operatorem lub znakiem "=".

## Wejście:

S – stos liczb zmiennoprzecinkowych

## Wyjście:

Wartość wyrażenia ONP na szczycie stosu S

## Elementy pomocnicze:

e – przechowuje odczytaną informację z wejścia jako łańcuch tekstowy

$v_1, v_2$  – przechowują argumenty operacji

## Lista kroków:

- |  |   |
|--|---|
| K01: <b>Czytaj</b> $e$   | <i>// odczytujemy kolejne elementy wyrażenia ONP</i>    |
| K02: <b>Jeśli</b> $e = "="$ , <b>to zakończ</b>  | <i>// znak = kończy wyrażenie ONP</i>                   |
| K03: <b>Jeśli</b> $e$ jest liczbą, <b>to idź do</b> K09                                    | <i>// liczby umieszczamy na stosie</i>                  |
| K04: $v_2 \leftarrow S.top()$<br>$S.pop()$   | <i>// pobieramy ze stosu argumenty operacji</i>         |
| K05: $v_1 \leftarrow S.top()$<br>$S.pop()$   |   |
| K06: Wykonaj operację na $v_1$ i $v_2$ zgodnie z zawartością $e$ .<br>Wynik umieść w $v_1$ | <i>// wykonujemy obliczenia zgodnie ze zn operatora</i> |
| K07: $S.push(v_1)$   | <i>// wynik trafia na stos</i>                          |
| K08: <b>Idź do</b> K01   | <i>// kontynuujemy przetwarzanie wyrażenia</i>          |
| K09: <b>Przekształć</b> $e$ na liczbę w $v_1$  |   |
| K10: <b>Idź do</b> K07   | <i>// liczbę umieszczamy na stosie</i>                  |

## Zadanie \*. Sortowanie

*Wybierz jeden algorytm sortowania, który nie został zaimplementowany podczas zajęć i na podstawie opisu algorytmu znalezionej w internecie, napisz ciało funkcji sortującej.*

## Zadanie \*\*. Wyszukiwanie binarne

*Zaimplementuj algorytm wyszukiwania binarnego.*