

Zestaw zadań konkursowych
XIX Regionalnego Konkursu Informatycznego

Instytut Informatyki UwB

30.03.2019 r.

Uwaga:

Wersje źródłowe i skompilowane programy mają być umieszczone na Pulpicie w katalogu o nazwie „konkurs”. Nazwy plików: z1.cpp, z1.exe – program będący rozwiązaniem zadania 1, z2.cpp, z2.exe – zadania 2 itd.

Zadanie 1: „Kochany” Szef

Masz „kochanego” nowego Szefa. Tzn. Ty jesteś nowy w Firmie, Szef jest już tak przyklejony do swojego stołka, że rzadko z niego wstaje. Twój poprzednik, po wielu perypetiach, postanowił ulotnić się z Firmy. Jednak pozostały po nim zadania, które mają być wykonane natychmiast, „na wczoraj”. Siedzisz przy komputerze poprzednika, Szef wspomniał tylko, że masz napisać cztery programy, dane do nich znajdziesz na wirtualnej maszynie w plikach o zaskakujących nazwach: zad2.pdf, zad3.pdf, zad4.pdf i zad5.pdf. Nic prostszego! Firma pustoszeje, Szef już się ulotnił (stołek chyba jednak pozostał). Włączasz wirtualną maszynę i ... Houston, mamy problem! System chce hasło!!! Ale jakie? Szef nieosiągalny, nikt nie zna hasła do tej maszyny, a czasu coraz mniej... Rusz głową, może jednak można to jakoś ominąć? Masz pół godziny na próby, potem „dobre duszki” Ci pomogą, ale, niestety, w oczach Szefa będziesz miał minus.

Teraz pozostało tylko odnaleźć pliki z zadaniami – Twój poprzednik gdzieś je wrzucił, tylko gdzie?

UWAGA:

Programy proszę pisać z wykorzystaniem oprogramowania w systemie natywnym (nie wirtualnym)

Zadanie 2: Nowy protokół – ciąg dalszy

Twój poprzednik miał okazję zaimplementować algorytm kodowania danych zgodnie z nowym protokołem. To jednak dopiero pierwszy krok. Teraz osoba odbierająca dane musi umieć je odczytać. Do tego potrzebny będzie dekodery. Napisz program, który przetworzy w odpowiedni sposób zakodowany plik wejściowy i wyłuska z niego wiadomość tekstową.

Specyfikacja protokołu:

1. W pierwszym bajcie znajduje się liczba informująca o długości przesyłanych danych:
 - jeśli jest mniejsza niż 126, informuje o konkretnej ilości danych,
 - jeśli jest ona równa 126, oznacza to, że należy odczytać kolejne 2 bajty - na nich znajduje się właściwa ilość,
 - jeśli w pierwszym bajcie jest liczba 127, oznacza to, że w celu pobrania długości danych należy odczytać kolejne 8 bajtów.
2. Po informacji o długości danych znajdują się cztery bajty składające się na maskę potrzebną do odszyfrowania wiadomości.
3. Na końcu jest właściwa wiadomość zgodna z długością podaną na początku pakietu. Każda litera jest zaszyfrowana z wykorzystaniem maski i operacji XOR. Jeśli bajty w masce się skończą, należy wrócić do jej początku. Czyli np. drugi bajt wiadomości będzie poddany operacji XOR z drugim bajtem maski, a piąty bajt wiadomości z pierwszym bajtem maski. Dla uproszczenia problemu zakładamy, że wiadomość nie będzie zawierała polskich znaków, a jedynie te występujące w tablicy ASCII.

Dane wejściowe

Wiadomość zakodowana zgodnie z naszym protokołem.

Dane wyjściowe

Odkodowana wiadomość.

Przykład:

Dane wejściowe

```
bkj,?.?..D..X...VW..MJ..U...MIK.BO...K>IE...L..VG.JCJ..HA..BWK.^T.JYT...E[K.M  
J..IDK.M]...J.JXK.J\A..MM.D
```

Widok w edytorze HEX:

```
00000000 62 6B 6A 2C 2E 3F 05 0C 44 0E 19 58 0E 1B 18 56  bkj,?.?..D..X...V  
00000010 57 00 06 4D 4A 04 1D 55 0E 08 03 4D 49 4B 10 42  W..MJ..U...MIK.B  
00000020 4F 00 05 5B 00 4B 3E 49 45 18 1E 0C 4C 0E 0E 56  O..[.K>IE...L..V  
00000030 47 0E 4A 43 4A 00 05 48 41 1C 0B 42 57 4B 1A 5E  G.JCJ..HA..BWK.^  
00000040 54 12 4A 59 54 12 09 45 5B 4B 10 4D 4A 0A 04 49  T.JYT...E[K.MJ..I  
00000050 44 4B 07 4D 5D 00 03 0C 4A 04 4A 58 4B 01 4A 5C  DK.M]...J.JXK.J\  
00000060 41 18 1E 4D 4D 02 44  A..MM.D
```

Dane wyjściowe

To jest przykładowy ciąg znaków. Tekst będzie odkodowany przy użyciu zadanej maski do tej postaci.

Zadanie 3: Kalkulator

W Firmie potrzebny jest kalkulator przetwarzający całe wyrażenia arytmetyczne. Niestety, nikt nie posiada klawiatury z przyciskami odpowiedzialnymi za wpisywanie nawiasów. Ponieważ kupno nowych klawiatur wprowadziło Szefa w stan przedzawałowy („ludzie, przecież to straszne koszty!!!”), Ty musisz napisać program tak, żeby nawiasy nie były potrzebne. Jest to wykonalne, jeśli zamiast zapisu infiksowego wejście będzie zgodne z zapisem postfiksowym (znak wykonywanej operacji umieszczany jest po operandach). Twoim zadaniem jest napisanie programu, który przetworzy takie wyrażenia arytmetyczne i na wyjściu wypisze ich wynik.

Zakładamy, że na wejściu mogą pojawiać się liczby rzeczywiste oraz znaki:

- '+' - dodawanie,
- '-' - odejmowanie,
- '/' - dzielenie,
- '*' - mnożenie,
- '^' - potęgowanie.

Wszystkie obsługiwane operacje są dwuargumentowe oraz zakładamy, że wejście będzie zawsze poprawnie skonstruowane (nie wystąpi np. dzielenie przez 0).

Dane wejściowe

Wyrażenie arytmetyczne zapisane postfiksowo.

Dane wyjściowe

Wynik wyrażenia arytmetycznego w postaci liczby całkowitej lub rzeczywistej.

Przykład:

Dane wejściowe

3 15 2 * / 10 + 2 ^

Dane wyjściowe

102.01

Zadanie 4: Perfekcyjne grupy

Szef wpadł na pomysł stworzenia systemu pozwalającego na, jego zdaniem, jak najbardziej rozsądny podział osób na grupy. Uznał, że w każdej grupie powinna być połowa najlepszych i najgorszych pracowników wg listy rankingowej. Na liście rankingowej osoby z najwyższą średnią ocen, które wystawia specjalna Komisja (Szef, jego Żona, dzieci, rodzice itd.) są najwyższe. W przypadku grup niepodzielnych bez reszty przez 2 (np. grupy 3 i 5 osobowe), większość grupy mają stanowić osoby z najwyższą średnią ocen. Napisz program dzielący osoby na grupy wg tych kryteriów.

Dane wejściowe

Pierwsza linia zawiera licznosc grup wieksza niz 1. Kolejne linie zawieraja unikalny, liczbowy numer pracownika/pracownicy oraz jego/jej srednia ocen.

Dane wyjściowe

Kazda linia danych wyjsciowych zawiera oddzielone spacja unikalne numery pracownikow z poszczegolnych grup uporzadkowane w kolejnosci rosnacej.

Przykład:

Dane wejściowe

```
2
1 3.4
2 4.0
5 4.8
3 2.4
4 3.0
6 4.5
```

Dane wyjściowe

```
3 5
4 6
1 2
```

Zadanie 5: Zarządzanie czasem

Żona Szefa ma tak mnóstwo obowiązków, że sama zaczyna się gubić co na kiedy ma wykonać. Pomóż jej zaplanować dzień biorąc pod uwagę wagę zadania oraz termin jego wykonania.

Dane wejściowe

Dane wejściowe złożone są z dwóch części oddzielonych trzema kropkami w nowej linii. Pierwsza część składa się z linii opisujących ramy czasowe do planowania aktywności. W każdej linii kolejno:

- data w formacie rok-miesiąc-dzień (dni i miesiące zapisane są za pomocą dwóch cyfr, a rok za pomocą 4 cyfr),
- godzina rozpoczęcia okna czasowego do planowania zadań na ten dzień,
- godzina zakończenia okna czasowego do planowania zadań na ten dzień.

Druga część składa się z linii opisujących aktywności do wykonania. W każdej linii:

- unikalny numer aktywności,
- ważność (priorytet) aktywności wyrażonej cyfrą większą niż 1 (1 - najwyższy priorytet),
- czas trwania aktywności w godzinach (cyfra większa niż 1),
- data wykonania aktywności w formacie rok-miesiąc-dzień (dni i miesiące zapisane są za pomocą dwóch cyfr, a rok za pomocą 4 cyfr).

Poszczególne wartości w liniach oddzielone są spacjami. Zakładamy, że nie można podać więcej aktywności niż długość trwania dnia, a także, że daty pokrywają się w obu częściach.

Dane wyjściowe

Program wypisuje datę w formacie rok-miesiąc-dzień (dni i miesiące zapisane są za pomocą dwóch cyfr, a rok za pomocą 4 cyfr), a w kolejnych liniach zaplanowane na ten dzień aktywności:

- godzinę rozpoczęcia aktywności,
- godzinę zakończenia aktywności,
- unikalny numer aktywności.

Wartości oddzielone są znakiem spacji. Poszczególne daty wypisywane są w kolejności od najstarszej do najnowszej.

Kolejność przydzielania zadań do dni:

1. priorytet zadania,
2. czas trwania aktywności,
3. numer aktywności.

Aktywności przydzielane są od godziny rozpoczęcia okna czasowego.

W przypadku, gdy aktywność zaplanowana na dany dzień nie mieści się w zaplanowanym oknie czasowym, na końcu linii przy tej aktywności program dopisuje znak *.

Przykład:

Dane wejściowe

2019-03-30 10 14

2019-03-31 9 14

2019-04-01 8 15

...

1 1 2 2019-03-30
2 2 3 2019-04-01
4 1 1 2019-03-30
3 2 4 2019-03-30
5 1 2 2019-03-31

Dane wyjściowe

2019-03-30
10 11 4
11 13 1
13 17 3*
2019-03-31
9 11 5
2019-04-01
8 11 2