

ad-wdi-k1a, wtorek, 4 listopada 2014

1. Proszę napisać program, który wczytuje wprowadzany z klawiatury ciąg liczb naturalnych zakończonych zerem stanowiącym wyłącznie znacznik końca danych.

Program powinien wypisać 10 co do wielkości wartość, jaka wystąpiła w ciągu.

Na przykład dla ciągu: 1,2,3,2,3,4,5,6,7,8,9,9,11,12,13,0 powinna zostać wypisana liczba 3.

Można założyć, że w ciągu znajduje się wystarczająca liczba elementów.

2. Dane są dwie tablice mogące pomieścić taką samą liczbę elementów:

```
int t1[N][N];
```

```
int t2[M];          // M = N*N
```

W każdym wierszu tablicy t1 znajdują się uporządkowane niemalejąco (w obrębie wiersza) liczby naturalne. Proszę napisać fragment programu przepisujący wszystkie liczby z tablicy t1 do t2, tak aby liczby w tablicy t2 były uporządkowane niemalejąco.

ad-wdi-k1b, środa, 5 listopada 2014

1. Proszę napisać program, który wczytuje wprowadzany z klawiatury ciąg liczb naturalnych zakończonych zerem stanowiącym wyłącznie znacznik końca danych.

Program powinien wypisać te elementy ciągu które są równe średniej arytmetycznej z 4 najbliższych sąsiadów. Na przykład dla ciągu: 2,3,2,7,1,2,4,8,5,2,2,4,3,9,5,4,0 powinny zostać wypisane podkreślone liczby. Można założyć, że w ciągu znajduje się co najmniej 5 elementów.

2. Dane są dwie tablice mogące pomieścić taką samą liczbę elementów:

```
int t1[10][N];
```

```
int t2[M];          // M = 10*N
```

W każdym wierszu tablicy t1 znajdują się uporządkowane rosnąco (w obrębie wiersza) liczby naturalne.

Proszę napisać fragment programu przepisujący wszystkie singletony (liczby występujące dokładnie raz) z tablicy t1 do t2, tak aby liczby w tablicy t2 były uporządkowane rosnąco. Pozostałe elementy tablicy t2 powinny zawierać zera.

Uwagi:

- można założyć poprawność danych wejściowych;
- oceniane będą: czytelność, poprawność i efektywność rozwiązań;
- za każde zadanie można otrzymać maksymalnie 5 pkt;
- czas na rozwiązanie zadań wynosi 30 min.

ad-wdi-k2a, wtorek, 9 grudnia 2014

1. Na szachownicy o wymiarach 201 wierszy i 201 kolumn umieszczamy 100 króli szachowych. Proszę napisać program, który wczytuje z klawiatury położenia 100 króli (wiersz, kolumna), odnajduje dwa króle jednakowo odległe od środka szachownicy i wypisuje ich pozycję (wiersz, kolumna). W przypadku gdy żadna para króli nie spełnia warunku program kończy się stosownym komunikatem. Odległość króla od środka to liczba jego ruchów, które musi wykonać aby dotrzeć do środka szachownicy.

Uwagi:

1. Król może przesunąć się na dowolne z 8 sąsiednich pól
2. Można założyć, że dane wprowadzone z klawiatury będą poprawne
3. Licząc drogę króla zakładamy, że szachownica jest pusta

2. Wyrazy budowane są z liter a..z. Dwa wyrazy „ważą” tyle samo jeżeli: mają tę samą liczbę samogłosek oraz sumy kodów ascii liter z których są zbudowane są identyczne, na przykład „ula” -> 117 108 97 oraz „exe” 101 120 101. Proszę napisać funkcję bool wyraz(string s1, string s2), która sprawdza czy jest możliwe zbudowanie wyrazu z podzbioru liter zawartych w s2 ważącego tyle co wyraz s1. Dodatkowo funkcja powinna wypisać znaleziony wyraz.

ad-wdi-k2b, środa, 10 grudnia 2014

1. Na szachownicy o wymiarach 201 wierszy i 201 kolumn umieszczono pewną liczbę wież szachowych tak, że każde z pól na jest szachowane. Przyszedł zły człowiek i zmienił położenie jednej z wież na szachownicy, tak że nie wszystkie pola są szachowane. Proszę zaproponować funkcję, która znajdzie przeniesienie jednej wieży tak aby ponownie wszystkie pola były szachowane. Do funkcji przekazujemy tablicę bool t[201][201] z układem wież po zmianie, funkcja powinna wyznaczyć i zwrócić dwa pola (wiersz, kolumna) – skąd , dokąd należy przenieść wieżę.

2. Dane są dwie liczby naturalne z których budujemy trzecią liczbę. W budowanej liczbie muszą wystąpić wszystkie cyfry występujące w liczbach wejściowych. Wzajemna kolejność cyfr każdej z liczb wejściowych musi być zachowana. Na przykład mając liczby 123 i 75 możemy zbudować liczby 12375, 17523, 75123, 17253, itd. Proszę napisać funkcję która wyznaczy ile liczb pierwszych można zbudować z dwóch zadanych liczb.

ad-wdi-k3a, wtorek, 20 stycznia 2015

1. Dana jest tablica wypełniona liczbami naturalnymi:

```
const int N=1000; int t[N][N];
```

Proszę napisać funkcję, która poszukuje w tablicy kwadratu o liczbie pól będącej liczbą nieparzystą większą od 1, którego iloczyn 4 pól narożnych wynosi k. Do funkcji należy przekazać tablicę i wartość k. Funkcja powinna zwrócić informacje czy udało się znaleźć kwadrat oraz współrzędne (wiersz, kolumna) środka kwadratu.

2. Dane są dwa jednokierunkowe łańcuchy odsyłaczowe (listy) zbudowane z elementów:

```
struct node { int w; node* next; };
```

Każdy łańcuch zawiera niepowtarzające się liczby naturalne. W pierwszym łańcuchu liczby są posortowane rosnąco, a w drugim nie. Proszę napisać funkcję usuwającą z obu łańcuchów liczby występujące w obu łańcuchach. Do funkcji należy przekazać wskazania na oba łańcuchy, funkcja powinna zwrócić łączną liczbę usuniętych elementów.

Uwagi:

1. Oceniane będą czytelność, poprawność, efektywność rozwiązania
2. Za zadanie można otrzymać maksymalnie 5 pkt.
3. Czas na rozwiązanie obu zadań wynosi 40 min.
4. W zadaniu 2 rozważyć użycie wartownika.

ad-wdi-k3b, środa, 21 stycznia 2015

-

1. Dana jest tablica wypełniona liczbami naturalnymi:

```
const int N=1000;  
int t[N][N];
```

Proszę napisać funkcję, która w poszukuje w tablicy najdłuższego ciągu geometrycznego leżącego ukośnie w kierunku prawo-dół, liczącego co najmniej 3 elementy. Do funkcji należy przekazać tablicę. Funkcja powinna zwrócić informacje czy udało się znaleźć taki ciąg oraz długość tego ciągu.

Uwagi:

1. Oceniane będą czytelność, poprawność, efektywność rozwiązania
2. Za zadanie można otrzymać maksymalnie 5 pkt.
3. Czas na rozwiązanie obu zadań wynosi 40 min.
4. W zadaniu 2 rozważyć użycie wartownika.

2. Dane są dwa jednokierunkowe łańcuchy odsyłaczowe (listy) zbudowane z elementów:

```
struct node { int w; node* next; };
```

Każdy łańcuch zawiera niepowtarzające się liczby naturalne. W obu łańcuchach liczby są posortowane rosnąco.

Proszę napisać funkcję usuwającą z każdego łańcucha liczby nie występujące w drugim. Do funkcji należy przekazać wskazania na oba łańcuchy, funkcja powinna zwrócić łączną liczbę usuniętych elementów.