ad-wdi kolokwium 1, czwartek, 28 stycznia 2010 Nazwisko Imię

Zad. 1 Dana jest duża tablica typu *tab=array[1..n] of integer*. Proszę napisać **funkcję**, która zwraca informację, czy w tablicy zachodzi następujący warunek: „wszystkie elementy, których indeks jest elementem ciągu Fibonacciego są liczbami złożonymi, a wśród pozostałych przynajmniej jedna jest liczbą pierwszą".

Uwagi:

- Czas na rozwiązanie zadania wynosi 25 minut, za zadanie można otrzymać 5 punktów.
- Oceniane będą: przejrzystość i czytelność kodu oraz efektywność rozwiązania.

Zad. 2 Dany jest zbiór n liczb naturalnych umieszczony w tablicy typu *tab=array[1..n] of integer*. Proszę napisać **funkcję**, która zwraca informację, czy jest możliwy podział zbioru n liczb na trzy podzbiory, tak aby w każdym podzbiorze, łączna liczba jedynek użyta do zapisu elementów tego podzbioru w systemie dwójkowym była jednakowa. Na przykład:

{2,3,5,7,15} -> true, bo podzbiory {2,7} {3,5} {15} wymagają użycia 4 jedynek,
{5,7,15} -> false, podział nie istnieje.

Uwagi:

- Zawartość tablicy wejściowej nie może ulec zmianie.
- Czas na rozwiązanie zadania wynosi 25 minut, za zadanie można otrzymać 5 punktów.
- Oceniane będą: przejrzystość i czytelność kodu oraz efektywność rozwiązania.
- Dodatkowe 2 pkt. można otrzymać, jeżeli funkcja zamiast informacji logicznej, w efektywny sposób policzy i zwróci liczbę istotnie różnych podziałów zbioru na podzbiory. Na przykład:
  {2,3,5,7,11,15} -> 2, bo {2,15} {3,7} {5,11} albo {2,15} {3,11} {5,7}

Zad. 3 Dany jest łańcuch zbudowany w oparciu o elementy typu:

```
pnode = ^ node;
node = record
        val : integer;
        next : pnode;
end;
```

Kolejne elementy łańcucha o zwiększającej się wartości pola *val* nazywamy podłańcuchem rosnącym. Proszę napisać **procedurę**, która usuwa z łańcucha wejściowego najdłuższy podłańcuch rosnący. Warunkiem usunięcia jest istnienie w łańcuchu dokładnie jednego najdłuższego podłańcucha rosnącego.

Uwagi:

- Czas na rozwiązanie zadania wynosi 25 minut, za zadanie można otrzymać 5 punktów.
- Oceniane będą: przejrzystość i czytelność kodu oraz efektywność rozwiązania.

ad-wdi kolokwium 2, wtorek, 9 lutego 2010 Nazwisko Imię

Zad. 1 Dana jest duża tablica typu *tab=array[1..n] of integer*. Proszę napisać **funkcję**, która zwraca sumę cyfr elementów tablicy zapisanych w systemie o podstawie 7, których indeks nie jest liczbą pierwszą. Wskazówka: rozważyć metodę sita.

Uwagi:

- Zawartość tablicy wejściowej nie może ulec zmianie.
- Czas na rozwiązanie zadania wynosi 25 minut, za zadanie można otrzymać 5 punktów.
- Oceniane będą: przejrzystość i czytelność kodu oraz efektywność rozwiązania.

Zad. 2 Dana jest tablica typu *tab=array[1..n] of integer*. Proszę napisać **funkcję**, która znajdzie najmniejszy (w sensie liczebności) podzbiór elementów tablicy, dla którego suma elementów jest równa sumie indeksów tych elementów. Do funkcji należy przekazać tablicę, funkcja powinna zwrócić sumę elementów znalezionego podzbioru. Na przykład dla tablicy: [7, 3, 5, 11, 2] rozwiązaniem jest liczba 10.

Uwagi:

- Zawartość tablicy wejściowej nie może ulec zmianie.
- Czas na rozwiązanie zadania wynosi 25 minut, za zadanie można otrzymać 5 punktów.
- Oceniane będą: przejrzystość i czytelność kodu oraz efektywność rozwiązania.

Zad. 3 Dany jest łańcuch zbudowany w oparciu o elementy typu:

```
pnode = ^ node;
node = record
        val : integer;
        next : pnode;
end;
```

Dane są dwa niepuste łańcuchy, z których każdy zawiera niepowtarzające się elementy. Elementy w pierwszym łańcuchu są uporządkowane rosnąco, w drugim elementy występują w przypadkowej kolejności. Proszę napisać **procedurę**, która z dwóch takich łańcuchów stworzy jeden, w którym uporządkowane elementy będą stanowić sumę mnogościową elementów z łańcuchów wejściowych. Do procedury należy przekazać wskazania na oba łańcuchy. Na przykład dla łańcuchów:

2 -> 3 -> 5 ->7-> 11

8 -> 2 -> 7 -> 4

powinien pozostać łańcuch:

2 -> 3 -> 4 -> 5 ->7-> 8 -> 11

Uwagi:

- Czas na rozwiązanie zadania wynosi 25 minut, za zadanie można otrzymać 5 punktów.
- Oceniane będą: przejrzystość i czytelność kodu oraz efektywność rozwiązania.

ad-wdi kolokwium 3, poniedziałek, 22 lutego 2010 Nazwisko Imię

Zad. 1 Dana jest duża tablica typu *tab=array [1..max, 1..max] of integer* wypełniona liczbami naturalnymi. Proszę napisać **funkcję**, która sprawdza czy w tablicy istnieją dwa elementy odległe o jeden ruch skoczka szachowego, których wartości są liczbami względnie pierwszymi.

Uwagi:

- Zawartość tablicy wejściowej nie może ulec zmianie.
- Czas na rozwiązanie zadania wynosi 25 minut, za zadanie można otrzymać 5 punktów.
- Oceniane będą: przejrzystość i czytelność kodu oraz efektywność rozwiązania.

Zad. 2 Do budowy liczby naturalnej reprezentowanej w systemie dwójkowym możemy użyć A cyfr 1 oraz B cyfr 0, gdzie A,B>0. Proszę napisać **funkcję**, która dla zadanych parametrów A i B zwraca ilość wszystkich możliwych do zbudowania liczb, takich że pierwsza cyfra w systemie dwójkowym (najstarszy bit) jest równa 1, a zbudowana liczba jest złożona.

Na przykład dla A=2, B=3 ilość liczb wynosi 3, są to $10010_{(2)}$ $10100_{(2)}$ $11000_{(2)}$

Uwagi:

- Czas na rozwiązanie zadania wynosi 25 minut, za zadanie można otrzymać 5 punktów.
- Oceniane będą: przejrzystość i czytelność kodu oraz efektywność rozwiązania.

Zad. 3 Dany jest łańcuch, zawierający liczby naturalne, zbudowany w oparciu o elementy typu:

```
pnode = ^ node;
node = record
        val : integer;
        next : pnode;
end;
```

Proszę napisać **funkcję**, która rozdziela elementy łańcucha wejściowego do 10 łańcuchów, według ostatniej cyfry pola *val*. W drugim kroku powstałe łańcuchy należy połączyć w jeden łańcuch, który jest posortowany niemalejąco według ostatniej cyfry. Do funkcji należy przekazać wskazanie na łańcuch wejściowy, funkcja powinna zwrócić wskazanie na powstały łańcuch.

Na przykład dla łańcucha:

2 -> 3 -> 5 ->7-> 10-> 11->23->13->17->24

Łańcuch wyjściowy może wyglądać następująco:

10 -> 11 -> 2 -> 13 -> 3 -> 24 -> 5 -> 17 -> 7

Uwagi:

- Wskazówka: należy skorzystać z typu: *tab_wsk = array [0..9] of pnode;*
- Czas na rozwiązanie zadania wynosi 25 minut, za zadanie można otrzymać 5 punktów.
- Oceniane będą: przejrzystość i czytelność kodu oraz efektywność rozwiązania.