

1. [7] Dana jest następująca deklaracja:

```
struct elem { int val; struct elem *next; }
```

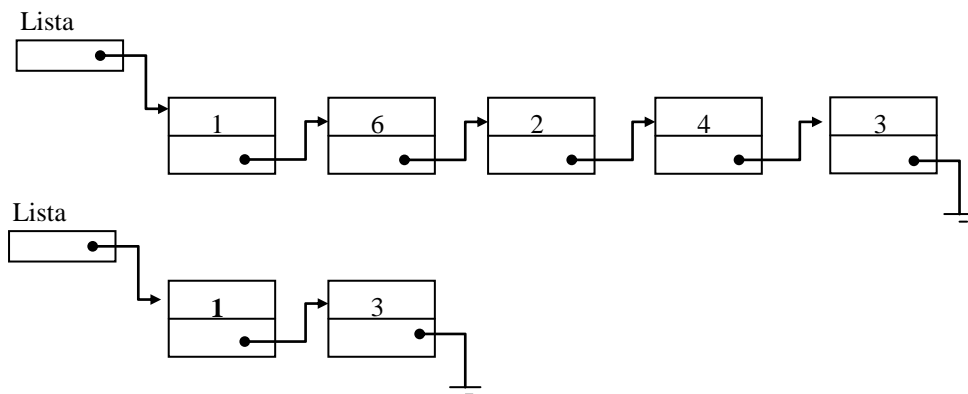
```
class ListItem:  
    def __init__(self,value): self.val = value; self.next =  
    None
```

Wersja 1 [7]: Napisz funkcję `usuwaj`, która z listy podanej jako argument usunie wszystkie elementy o parzystej wartości `val` i zwróci wynikową listę jako wartość. Np. dla listy składającej się z elementów 1, 6, 2, 4, 3 wywołanie `usuwaj(lista)` powinno dać w wyniku listę 1, 3.

Nagłówek funkcji w języku C powinien mieć postać

```
struct elem *usuwaj(struct elem *lista)
```

a w Pythonie `def usuwaj(lista)`



Rys. Górny rysunek przedstawia sytuację przed wykonaniem funkcji, a dolny – po wykonaniu.

Wersja 2 [4]: Napisz funkcję `monoton`, która dla listy podanej jako argument zwraca 1 gdy elementy listy tworzą ciąg niemalejący, a 0 w przeciwnym przypadku. Np. dla listy 1, 6, 2, 4, 3 z rysunku wartość powinna wynieść 0, a dla ciągu 1, 2, 4, 7 wartość to 1.

Imię i nazwisko:
Indeks:.....

WdI, Kolokwium nr 2

2. [6] Dany jest program P (pomijamy nagłówki i instrukcje wejścia):

Wersja C: { int res=0, i=0; while (i<n) { res = res + i; i = i + 1; } return res; }	Wersja Python: res=0 i=0 while i<n: res = res + i i = i + 1 return res
-----------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------

i fragment jego formalnej specyfikacji:

{ n - dodatnia liczba naturalna } P { res=..... }

Twoje zadanie:

- Uzupełnij warunek końcowy powyższej specyfikacji, tak aby zwracana liczba res była jednoznacznie określona.
- Sformułuj niezmiennik I pętli while, który umożliwi wykazanie poprawności Twojej specyfikacji
- Uzasadnij, że jeśli spełniony jest warunek $I \wedge (p < n)$ przed obrotem pętli while, to po jednym obrocie pętli nadal spełniony jest niezmiennik I.

Twoje odpowiedzi:

Imię i nazwisko:
Indeks:.....

3. [5] Dane są następujące funkcje:

(01) int sm(int i, int j, int a[])	def sm(i, j, a):
(02) { if (i==j) return a[i];	if i==j: return a[i]
(03) s = (i+j) / 2;	s = (i+j) / 2
(04) s1 = sm(i, s, a);	s1 = sm(i, s, a)
(05) s2 = sm(s+1, j, a);	s2 = sm(s+1, j, a)
(06) return s1*s2;	return s1*s2
(07) }	
(08)	
(09) int bsm(int n, int a[]){	def bsm(n, a):
(10) return sm(0, n-1, a);	return sm(0, n-1, a)
(11) }	

- a) [1] Podaj drzewo wywołań rekurencyjnych dla wywołania sm(10,19,a)
b) [2] Uzupełnij specyfikację funkcji bsm.

Specyfikacja

Wejście: n– liczba naturalna; a – tablica liczb naturalnych

Wyjście:

.....

.....

- c) [2] Podaj zależność rekurencyjną, która określa liczbę mnożeń wykonywanych w wierszu (06) przy uruchomieniu bsm(a, n) dla n postaci $n=2^p$, gdzie p jest liczbą naturalną.

Zależność rekurencyjna

$T(1) = 0$

$T(n) = \dots\dots\dots$ dla $n > 1$, n potęgą liczby 2

Podaj uzasadnienia do punktów b) i c).

Imię i nazwisko:
Indeks:.....

WdI, Kolokwium nr 2

4. [8] Napisz funkcję rozwiązującą następujący problem oraz oszacuj złożoność czasową i pamięciową swojego rozwiązania.

Wejście: n – dodatnia liczba naturalna

Wyjście:

Ciąg składający się ze wszystkich liczb naturalnych mniejszych od n , które można przedstawić jako sumę kwadratów dwóch dodatnich liczb naturalnych (niekoniecznie różnych).

Uwagi:

- wynik może zostać wypisany na ekranie, nie jest konieczne aby był zwracany jako wartość funkcji.
- maksymalną liczbę punktów można uzyskać tylko za rozwiązania działające w czasie $O(n)$.

Przykłady:

Dla $n=9$ w odpowiedzi powinny znaleźć się liczby 2, 5, 8 (gdyż $2=1^2+1^2$, $5=1^2+2^2$, $8=2^2+2^2$).
Liczby wynikowe mogą być wypisane w dowolnej kolejności, ale każda tylko jeden raz.

Wskazówka: możesz w tablicy rozmiaru n „zaznaczać” sumy uzyskane jako i^2+j^2 dla $1 \leq i, j \leq \sqrt{n}$

Imię i nazwisko:
Indeks:.....

WdI, Kolokwium nr 2