PRM kolokwium 2

A
imię, nazwisko
grupa studencka
Zad.1
Zad.2
Zad.3
suma

Czas pisania kolokwium **45 minut**

Do uzyskania w sumie **30 punktów**

Proszę odpowiadać na zadane pytania i rozwiązywać zadania według treści przekazanej na kartce. Zrozumienie treści zadania jest elementem rozwiązania. Dopuszczalne jest korzystanie z notatek i książek, żadne urządzenia elektroniczne nie są dozwolone. Punktacja jest określona przy numerze zadania.
Proszę uzupełnić dwa pierwsze pola tabelki obok. Zadań proszę nie przepisywać. Wszystkie oddawane kartki powinny być podpisane własnym imieniem i nazwiskiem. Proszę wyraźnie zaznaczyć odpowiedzi. Nie uzupełnienie tabelki powoduje ocenę 0 z kolokwium. Kartki nie podpisane nie będą sprawdzane.

Zad 1(15)

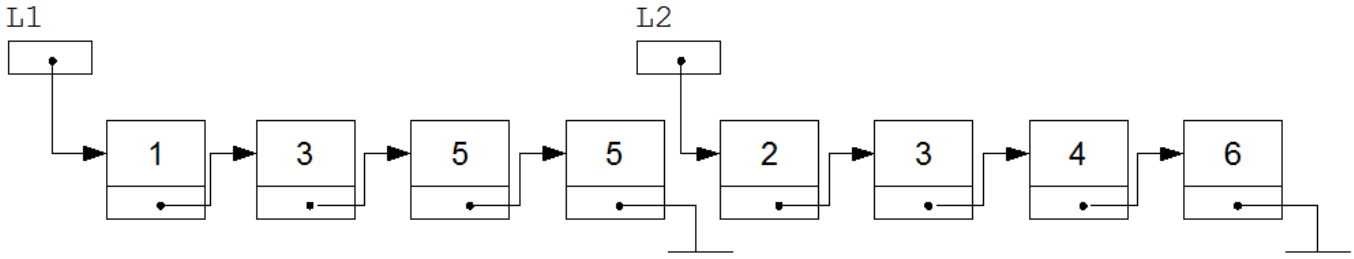
Załóżmy, że mamy listę liniową, której kostka składowa jest postaci

```
typedef struct w wezel;
struct w{
   int info;
   wezel *nast;
};
```

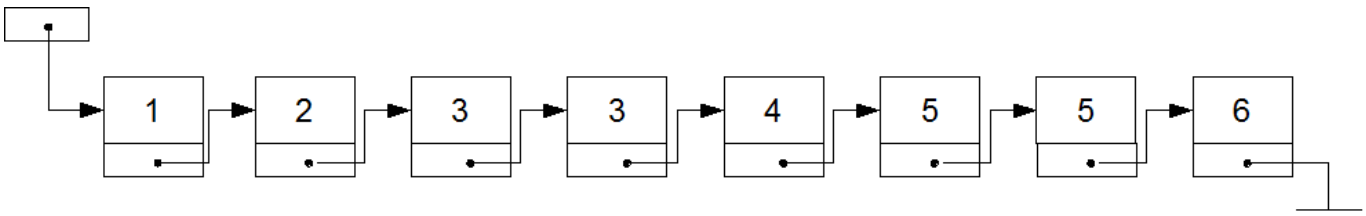
Pole `nast` w zawiera wskaźnik następnego elementu listy. Jeśli taki element nie istnieje, wówczas ma ono wartość `NULL`. Proszę napisać funkcję o nagłówku

```
wezel* zlacz (wezel *L1, wezel *L2);
```

która tworzy nową listę, posortowaną rosnąco według pola `info` i zwraca jako swój wynik punkt wejścia do nowej listy. Ta wynikowa lista jest wynikiem złączenia list `L1` i `L2`, z których każda jest posortowana rosnąco według pola `info`. Listy przekazane jako argumenty funkcji `zlacz` mają pozostać niezmienione. Można założyć, że istnieje funkcja `wezel* nowy (int info)` która tworzy węzeł o wartości pola `info` przekazanej jako argument wywołania.
Przykład wykonania



Wynik działania funkcji `zlacz`



PRM kolokwium 2

B
imię, nazwisko
grupa studencka
Zad.1
Zad.2
Zad.3
suma

Czas pisania kolokwium **45 minut**

Do uzyskania w sumie **30 punktów**

Proszę odpowiadać na zadane pytania i rozwiązywać zadania według treści przekazanej na kartce. Zrozumienie treści zadania jest elementem rozwiązania. Dopuszczalne jest korzystanie z notatek i książek, żadne urządzenia elektroniczne nie są dozwolone. Punktacja jest określona przy numerze zadania.
Proszę uzupełnić dwa pierwsze pola tabelki obok. Zadań proszę nie przepisywać. Wszystkie oddawane kartki powinny być podpisane własnym imieniem i nazwiskiem. Proszę wyraźnie zaznaczyć odpowiedzi. Nie uzupełnienie tabelki powoduje ocenę 0 z kolokwium. Kartki nie podpisane nie będą sprawdzane.

Zad 1(15)

Załóżmy, że mamy listę liniową, której kostka składowa jest postaci

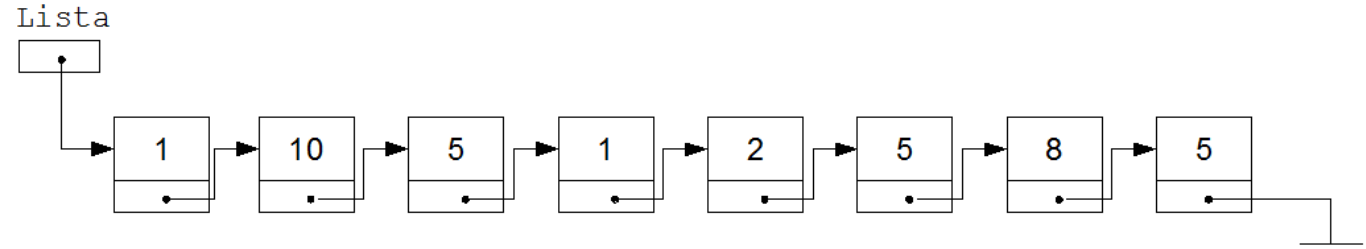
```
typedef struct w wezel;
struct w{
   int info;
   wezel *nast;
};
```

Pole `nast` w zawiera wskaźnik następnego elementu listy. Jeśli taki element nie istnieje, wówczas ma ono wartość `NULL`. Proszę napisać funkcję o nagłówku

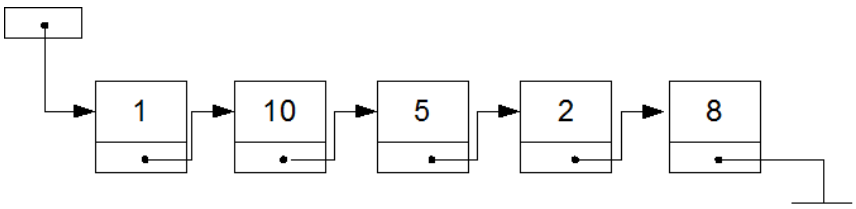
```
wezel* usunDup (wezel *Lista);
```

która tworzy nową listę i zwraca jako swój wynik punkt wejścia do niej. Ta wynikowa lista powstaje z listy `Lista` poprzez skopiowanie wszystkich elementów o unikalnych wartościach pola `info`. Innymi słowy, w liście wynikowej nie ma dwóch elementów o tej samej wartości pola `info` zatem oraz każda wartość `info` zawarta w liście `Lista` jest dokładnie raz zawarta w liści wynikowej. `Lista` przekazana jako argument funkcji `zlacz` ma pozostać niezmieniona. Można założyć, że istnieje funkcja `wezel* nowy (int info)` która tworzy węzeł o wartości pola `info` przekazanej jako argument wywołania.

Przykład wykonania



Wynik działania funkcji `usunDup`



Uwaga: dopuszczalna jest inna niż podana w przykładzie kolejność występowania wartości w liście wynikowej.

Zad 2(5)

Proszę naszkicować strukturę wskazań, która będzie wartością argumentu `w` po wykonaniu ostatniej instrukcji funkcji `fun` zakładając, że funkcja `fun` została wywołana następująco:

```
int main()
{  int **wsk;
/*tu jest pominiety kawalek kodu*/
   fun(5, &wsk);
/*tu jest ciag dalszy funkcji main*/
```

a każde wywołanie funkcji `malloc` zwróciło wartość różną od `NULL`.

```
void fun(int max, int*** w)
{
   int **pom;
   int i,j;

   pom=(int**)malloc(max*sizeof(int*));
   for (i=0; i<max; i++)
   {
      pom[i]=malloc((max-i)*sizeof(int));
      for (j=0; j<max-i; j++)
         pom[i][j]=i+j;
   }
   *w=pom; /*jak wygladaja teraz wskazania? */
}
```

Zad 3(10)

Jaki będzie stan standardowych strumieni wyjściowych w wyniku uruchomienia poniższego programu, jeśli program został uruchomiony z linii komend poleceniem "`./zad3 9`". Odpowiedź proszę krótko uzasadnić.

```
#include <stdio.h>

int fun(int x)
{
   static int y=1;
   int i;

   if (x>=2)
   {
      ++y;
      fprintf (stderr, "o");
      for (i=0; i<x%2; ++i)
         fprintf (stderr, "x");
      return fun(x/2);
   }
   else
      return y;
}

int main(int argc, char* argv[])
{
   int y;

   if (argc>1)
   {
       sscanf(argv[1],"%d",&y);
       printf("%d\n", fun(y));
   }
   return 0;
}
```

Zad 2(5)

Proszę naszkicować strukturę wskazań, która będzie wartością argumentu `w` po wykonaniu ostatniej instrukcji funkcji `fun` zakładając, że funkcja `fun` została wywołana następująco:

```
int main()
{  int **wsk;
/*tu jest pominiety kawalek kodu*/
   fun(5, &wsk);
/*tu jest ciag dalszy funkcji main*/
```

a każde wywołanie funkcji `malloc` zwróciło wartość różną od `NULL`

```
int fun(int max, int*** w)
{
   int **pom;
   int i,j;

   pom=(int**)malloc(max*sizeof(int*));
   for (i=0; i<max; i++)
   {
      pom[i]=malloc((i+1)*sizeof(int));
      for (j=0; j<=i; j++)
         pom[i][j]=i+j;
   }
   *w=pom; /*jak wygladaja teraz wskazania? */
}
```

Zad 3(10)

Jaki będzie stan standardowych strumieni wyjściowych w wyniku uruchomienia poniższego programu, jeśli program został uruchomiony z linii komend poleceniem "`./zad3 30`". Odpowiedź proszę krótko uzasadnić.

```
#include <stdio.h>

int fun(int x)
{
   static int y=1;
   int i;

   if (x>=3)
   {
      ++y;
      fprintf (stderr, "x");
      for (i=0; i<x%3; ++i)
         fprintf (stderr, "o");
      return fun(x/3);
   }
   else
      return y;
}

int main(int argc, char* argv[])
{
   int y;

   if (argc>1)
   {
       sscanf(argv[1],"%d",&y);
       printf("%d\n", fun(y));
   }
   return 0;
}
```