Cvičení 2

O P A K O V Á N Í

1. Uspořádané zřetězené seznamy

Co nejelegantnějším způsobem a s respektováním pravidel pro tvorbu programů zapište v programovacích jazycích Java a C (C++) následující funkce pro

- zařazení nové položky (nového prvku) do níže uvedených seznamů ve tvaru
  `void Zarad (P_element *ZAC_SEZ, *KON_SEZ; char *PRVEK) { ...` [à 2 body]
- zjištění přítomnosti prvku v seznamu ve tvaru
  `P_element Najdi (P_element ZAC_SEZ; char *PRVEK) { ...` [à 1 bod]
- vypuštění prvku ze seznamu ve tvaru
  `void Vypust (P_element *ZAC_SEZ, *KON_SEZ; char *PRVEK) { ...` [à 2 body]

takové, aby provedly příslušnou akci (operaci) nad

a) jednosměrně zřetězeným uspořádaným seznamem prvků obsahujících jako hodnoty libovolná křestní jména, přičemž deklarace prvku seznamu bude provedena následujícím způsobem:

```
typedef  struct element  *P_element;
typedef  struct element  {
            char *value;
            P_element next;
         } ELEM;

P_element  L = NULL;
```

b) obousměrně zřetězeným uspořádaným seznamem těchže prvků, přičemž deklarace prvku seznamu bude provedena:

```
typedef  struct element  *P_element;
typedef  struct element  {
            char *value;
            P_element next;
            P_element prev;
         } ELEM;

P_element  L = NULL;
```

Funkce `Najdi` nechť poskytne jako výstup hodnotu ukazatele na nalezený prvek v případě, že prvek byl nalezen, nebo hodnotu `NULL` v případě, že hledaný prvek v seznamu není.

2. Binární vyhledávací stromy

Mějte dán binární vyhledávací strom (dále jen BVS) deklarovaný následujícím způsobem:

```
typedef  struct element  *P_node;
typedef  struct element  {
            int value;
            P_node left_son, right_son;
         }  NODE;

P_node   BVS = NULL;
```

Co nejjednodušším způsobem a s respektováním pravidel pro tvorbu programů zapište v programovacích jazycích Java a C (C++):

- funkci `Search` pro vyhledání prvku o zadané celočíselné hodnotě v binárním vyhledávacím stromu – pokud je prvek ve stromu nalezen, funkce vrací nulovou hodnotu jako příznak nalezení prvku a v proměnné `S` hodnotu ukazatele na nalezený prvek, nebo hodnotu `-1` jako příznak nenalezení prvku a v proměnné `S` hodnotu ukazatele na místo v BVS, kam má být nově zařazovaný prvek o dané hodnotě vložen (funkce vrací ukazatel na místo, kam má být vložen ukazatel na nově zařazovaný prvek) s tím, že hlavička funkce bude mít tvar [à 2 body]

  `int Search (int ELEM; P_node *S) { ...` a

- funkci pro zařazení nového prvku do BVS na místo vyhledané funkcí SEARCH ve tvaru [à 1 bod]

  `void Insert (int ELEM; P_node *S) { ...`

Respektujte, že přístup do BVS je možný pouze prostřednictvím kořene stromu, tzn. že při volání procedur dosazujte za formální parametr `S` vždy symbolické jméno kořene stromu, do nějž budete daný prvek zařazovat, např.

```
HELP = BST;
if (Search (NODE_VAL, &HELP) == -1) Insert (NODE_VAL, *HELP);
```

Zvláště zdatní jedinci si mohou zadání dále ztížit tím, že napíší funkci pro zařazení prvku pro vyvážený BVS, tj. pro AVL strom.

3. Uspořádaný výpis obsahu tabulky

Máte dánu datovou strukturu tabulka podle níže uvedeného obrázku, která je startovní listinou soutěže ve víceboji a obsahuje dosažené výsledky. Tabulka je uspořádána vzestupně podle přidělených startovních čísel závodníků a pro vyhodnocení soutěže je rozhodující celkový počet získaných bodů (součet bodových hodnocení v jednotlivých disciplinách) uvedený v poslední kolonce tabulky:

	start. číslo	*příjmení a jméno*	*klubová příslušnost*	*body*
1	1	Bárta Michal	Družstvo A	6681
2	2	Cempírek Josef	Družstvo A	2457
3	3	Doležal Stanislav	Družstvo A	3211
4	4	Drobný Karel	Družstvo A	5423
5	5	Frantl Václav	Družstvo A	8162
6	6	Janda Vlastimil	Družstvo A	7189
7	7	Královec Bedřich	Družstvo A	5562
8	8	Lukáš Jaroslav	Družstvo A	6969
9	9	Prchal Stanislav	Družstvo A	4814
10	10	Pergler Miroslav	Družstvo A	8646
11	11	Burda Jan	Družstvo B	8567
12	12	Cudlín Josef	Družstvo B	4496
13	13	Dejmal Václav	Družstvo B	8209
14	14	Drahoňovský Karel	Družstvo B	6927
15	15	Géry Roman	Družstvo B	8189
16	16	Hála Vlastimil	Družstvo B	7323
17	17	Kukla Jan	Družstvo B	9512
18	18	Lemberk Stanislav	Družstvo B	6656
19	19	Vrána Karel	Družstvo B	8814
20	20	Ajgl Miroslav	Družstvo C	8764
21	21	Poštulka Lubomír	Družstvo C	7989
22	22	Řezníček Kamil	Družstvo C	9124
23	23	Tomášek Radek	Družstvo C	8579
24	24	Urbánek Karel	Družstvo C	4982
25	25	Zikmund Miroslav	Družstvo C	8356

Vaším úkolem je

1. nakreslit, jak byste nad danou tabulkou vytvořili ”nepřímý” binární vyhledávací strom (dále jen BVS), jímž by bylo možno vypsat výsledkovou listinu závodu (viz následující bod); klíčem pro zařazování prvků do stromu budiž počet dosažených bodů; [1 bod]
2. v programovacím jazyce C (C++) zapsat funkci `Vypis_Vysl_List`, která ze zobrazené tabulky vypíše výsledkovou listinu závodu řazenou od vítěze, který dosáhl nejvyššího počtu bodů, až po posledního závodníka s nejnižším počtem bodů dočasně vytvořeným lokálním ”nepřímým” BVS. [3 body]

4. Prohledávání obecného stromu

Mějte dán **B–strom 2. řádu** deklarovaný následujícím způsobem :

```
#define  M  2

typedef  struct element  *Bt_node;
typedef  struct element  {
           int      d;                   // pocet obsazenych pozic
           int      k[2*M];              // pole klicu
           Bt_node  p[2*M+1];            // odkazy na nasledovniky
         } NODE;

Bt_node    btree = NULL;
```

Zapište funkci `Search (t, key)`, která vyhledá prvek s klíčem `key` ve stromu `t` a jako funkční hodnotu vrátí ukazatel na nalezený prvek nebo `NULL`, pokud prvek ve stromu nebyl nalezen. [3 body]