

Registr obyvatel

Termín odevzdání:	14.04.2013 23:59:59
Pozdní odevzdání s penalizací:	12.05.2013 23:59:59 (Penále za pozdní odevzdání: 100.0000 %)
Hodnocení:	13.2000
Max. hodnocení:	10.0000 (bez bonusů)
Odevzdaná řešení:	6 / 20 Volné pokusy + 20 Penalizované pokusy (-2 % penalizace za každé odevzdání)
Nápovědy:	1 / 2 Volné nápovědy + 2 Penalizované nápovědy (-10 % penalizace za každou nápovědu)

Úkolem je realizovat třídu CRegister, která bude implementovat registr obyvatelstva.

Uvažovaný registr je specifický tím, že chceme ukládat všechny změny (všechny nahlášené změny bydliště daného člověka). Dále, pro účely archivace chceme ukládat kopie registru.

Úloha je směřována k procvičení pochopení mělké a hluboké kopie objektu. Má části povinné, dobrovolné a bonusové. Pro splnění povinné části úlohy postačuje vytvořit funkční třídu, která bude splňovat požadované rozhraní. Pro zvládnutí nepovinných a bonusových částí je potřeba rozmyslet ukládání dat tak, aby se při kopírování zbytečně neplýtvalo pamětí.

Úloha má procvičit práci s kopírováním objektů. Z tohoto důvodu jsou v úloze potlačené části standardní C++ knihovny, zejména STL a datový typ C++ string.

Požadovaná třída CRegister má následující rozhraní:

implicitní konstruktor
vytvoří prázdnou instanci registru,
kopírující konstruktor
vytvoří hlubokou kopii instance. Implementujte jej, pokud kompilátorem automaticky vytvořený kopírující konstruktor nevyhovuje (zkuste si odhadnout - bude potřeba?),
destruktor
uvolní prostředky alokované instancí.
operátor =
zkopíruje obsah jedné instance do druhé (hluboká kopie). Implementujte jej, pokud kompilátorem automaticky vytvořený operátor = nevyhoví,
Add(id,name,surname,date,street,city)
metoda přidá zadaného člověka do databáze a nastaví u něj adresu pobytu. Parametrem je 6 řetězců s následujícím významem:

- id udává jednoznačný identifikátor člověka. Jedná se o řetězec, můžete předpokládat, že je vždy ve formátu "XXXXXX/YYYY",
- name udává jméno člověka. Jméno může být libovolné, libovolně dlouhé.
- surname udává příjmení člověka. Může být libovolné, libovolně dlouhé.
- date udává datum počátku platnosti místa pobytu. Jedná se o řetězec, můžete předpokládat, že je vždy ve formátu "YYYY-MM-DD".
- street udává adresu pobytu, řetězec, libovolný obsah a délka.
- city udává adresu místa pobytu, řetězec, libovolný obsah a délka.

Metoda Add vrátí hodnotu true pro signalizaci úspěchu, false pro neúspěch (osoba se stejným id již byla zadána).

Resettle(id,date,street,city)

metoda přidá místo pobytu do záznamu daného člověka. Význam parametrů je stejný jako u metody Add. Metoda vrátí true pokud operace uspěje, false pokud selže (člověk se zadaným id neexistuje nebo daný člověk v ten samý den již dříve oznámil jinou adresu (tedy v jeden den se lze přestěhovat pouze jedenkrát). Pozor - informace o změně adresy nemusí být zpracovávány v pořadí vzrůstajícího času.

Print(stream,id)

metoda zobrazí záznam pro člověka zadaného id do zadaného výstupního proudu stream. Metoda vrátí true pro signalizaci úspěchu, false pro neúspěch (člověk se zadaným id neexistuje). Formát výstupu je uveden v ukázkách níže, adresy trvalého bydliště jsou v rostoucím pořadí.

Odevzdávejte soubor, který obsahuje implementovanou třídu CRegister a další Vaše podpůrné třídy. Třída musí splňovat veřejné rozhraní podle ukázky - pokud Vámi odevzdané řešení nebude obsahovat popsané rozhraní, dojde k chybě při kompilaci. Do třídy si ale můžete doplnit další metody (veřejné nebo i privátní) a členské proměnné. Odevzdávaný soubor musí obsahovat jak deklaraci třídy (popis rozhraní) tak i definice metod, konstruktoru a destruktoru. Je jedno, zda jsou metody implementované inline nebo odděleně. Odevzdávaný soubor nesmí obsahovat vkládání hlavičkových souborů a funkci main.

Funkce main a vkládání hlavičkových souborů může zůstat, ale pouze obalené direktivami podmíněného překladu jako v ukázce níže.

Pokud se rozhodnete řešit i nepovinné a bonusové části úlohy, můžete pro nalezení vhodné reprezentace využít následující pozorování:

- Vzniká mnoho kopií instance, které ale dále nejsou měněné. Tedy vyplatí se uvažovat realizaci, kde jsou data sdílená s originálem dokud nedojde k první modifikaci originálu nebo kopie.
- Změny v instanci jsou "malé", tedy velká část obsahu zůstane stejná jako v originálu. Vyplatí se tedy i uvažovat o sdílení nezměněných částí instance. Pokud byste si např. osoby realizovali jako třídy, dá se předpokládat, že mezi originálem a kopií se změní pouze několik osob.
- Řetězce jmen, příjmení, ale i ulice a města se opakují. V registru např. může být 10000 obyvatel, ale např. jen 500 unikátních příjmení. Vyplatí se uvažovat o sdílení jednotlivých řetězců nebo i o jejich deduplikaci (cílené vyhledávání stejných řetězců).

Požadované veřejné rozhraní třídy:

```
#ifndef __PROGTEST__
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <iostream>
using namespace std;
#endif /* __PROGTEST__ */

class CRegister
{
public:
    // default constructor
    // copy constructor
    // destructor
    // operator =

    bool                Add                ( const char    * id,
                                             const char    * name,
                                             const char    * surname,
                                             const char    * date,
                                             const char    * street,
                                             const char    * city );

    bool                Resettle           ( const char    * id,
                                             const char    * date,
                                             const char    * street,
                                             const char    * city );

    bool                Print              ( ostream      & os,
                                             const char    * id ) const;

private:
    // todo
};
```

Ukázka použití třídy:

```
bool    status;
char    lID[12], lDate[12], lName[50], lSurname[50], lStreet[50], lCity[50];
CRegister  a;
status = a . Add ( "123456/7890", "John", "Smith", "2000-01-01", "Main street", "Seattle" );
// status = true
status = a . Add ( "987654/3210", "Freddy", "Kruger", "2001-02-03", "Elm street", "Sacramento" );
// status = true
status = a . Resettle ( "123456/7890", "2003-05-12", "Elm street", "Atlanta" );
// status = true
status = a . Resettle ( "123456/7890", "2002-12-05", "Sunset boulevard", "Los Angeles" );
// status = true
status = a . Print ( cout, "123456/7890" );
/* status = true
----8<----8<----8<----8<----8<----8<----
123456/7890 John Smith
2000-01-01 Main street Seattle
2002-12-05 Sunset boulevard Los Angeles
2003-05-12 Elm street Atlanta
```

```

----8<----8<----8<----8<----8<----8<----
*/
status = a . Print ( cout, "987654/3210" );
/* status = true
----8<----8<----8<----8<----8<----8<----
987654/3210 Freddy Kruger
2001-02-03 Elm street Sacramento
----8<----8<----8<----8<----8<----8<----
*/
CRegister b ( a );
status = b . Resettle ( "987654/3210", "2008-04-12", "Elm street", "Cinccinati" );
// status = true
status = a . Resettle ( "987654/3210", "2007-02-11", "Elm street", "Indianapolis" );
// status = true
status = a . Print ( cout, "987654/3210" );
/* status = true
----8<----8<----8<----8<----8<----8<----
987654/3210 Freddy Kruger
2001-02-03 Elm street Sacramento
2007-02-11 Elm street Indianapolis
----8<----8<----8<----8<----8<----8<----
*/
status = b . Print ( cout, "987654/3210" );
/* status = true
----8<----8<----8<----8<----8<----8<----
987654/3210 Freddy Kruger
2001-02-03 Elm street Sacramento
2008-04-12 Elm street Cinccinati
----8<----8<----8<----8<----8<----8<----
*/
a = b;
status = a . Resettle ( "987654/3210", "2011-05-05", "Elm street", "Salt Lake City" );
// status = true
status = a . Print ( cout, "987654/3210" );
/* status = true
----8<----8<----8<----8<----8<----8<----
987654/3210 Freddy Kruger
2001-02-03 Elm street Sacramento
2008-04-12 Elm street Cinccinati
2011-05-05 Elm street Salt Lake City
----8<----8<----8<----8<----8<----8<----
*/
status = b . Print ( cout, "987654/3210" );
/* status = true
----8<----8<----8<----8<----8<----8<----
987654/3210 Freddy Kruger
2001-02-03 Elm street Sacramento
2008-04-12 Elm street Cinccinati
----8<----8<----8<----8<----8<----8<----
*/
status = b . Add ( "987654/3210", "Joe", "Lee", "2010-03-17", "Abbey road", "London" );
// status = false
status = a . Resettle ( "987654/3210", "2001-02-03", "Second street", "Milwaukee" );
// status = false
status = a . Print ( cout, "666666/6666" );
// status = false

CRegister c;
strncpy ( lID, "123456/7890", sizeof ( lID ) );
strncpy ( lName, "John", sizeof ( lName ) );
strncpy ( lSurname, "Smith", sizeof ( lSurname ) );
strncpy ( lDate, "2000-01-01", sizeof ( lDate ) );
strncpy ( lStreet, "Main street", sizeof ( lStreet ) );
strncpy ( lCity, "Seattle", sizeof ( lCity ) );
status = c . Add ( lID, lName, lSurname, lDate, lStreet, lCity );
// status = true
strncpy ( lID, "987654/3210", sizeof ( lID ) );
strncpy ( lName, "Freddy", sizeof ( lName ) );
strncpy ( lSurname, "Kruger", sizeof ( lSurname ) );

```

```

strncpy ( lDate, "2001-02-03", sizeof ( lDate ) );
strncpy ( lStreet, "Elm street", sizeof ( lStreet ) );
strncpy ( lCity, "Sacramento", sizeof ( lCity ) );
status = c . Add ( lID, lName, lSurname, lDate, lStreet, lCity );
// status = true
strncpy ( lID, "123456/7890", sizeof ( lID ) );
strncpy ( lDate, "2003-05-12", sizeof ( lDate ) );
strncpy ( lStreet, "Elm street", sizeof ( lStreet ) );
strncpy ( lCity, "Atlanta", sizeof ( lCity ) );
status = c . Resettle ( lID, lDate, lStreet, lCity );
// status = true
strncpy ( lID, "123456/7890", sizeof ( lID ) );
strncpy ( lDate, "2002-12-05", sizeof ( lDate ) );
strncpy ( lStreet, "Sunset boulevard", sizeof ( lStreet ) );
strncpy ( lCity, "Los Angeles", sizeof ( lCity ) );
status = c . Resettle ( lID, lDate, lStreet, lCity );
// status = true
status = c . Print ( cout, "123456/7890" );
/* status = true
----8<----8<----8<----8<----8<----8<----
123456/7890 John Smith
2000-01-01 Main street Seattle
2002-12-05 Sunset boulevard Los Angeles
2003-05-12 Elm street Atlanta
----8<----8<----8<----8<----8<----8<----
*/

```

☐ Referenční řešení

6	31.03.2013 23:54:41	Download
Stav odevzdání:		Ohodnoceno
Hodnocení:		13.2000
<ul style="list-style-type: none"> • Hodnotitel: automat <ul style="list-style-type: none"> ◦ Program zkompilován ◦ Test 'Zakladni test s parametry podle ukazky': Úspěch <ul style="list-style-type: none"> ▪ Dosáženo: 100.00 %, požadováno: 100.00 % ▪ Celková doba běhu: 0.000 s (limit: 3.000 s) ▪ Využití paměti: 12576 KiB (limit: 111017 KiB) ▪ Úspěch v závazném testu, hodnocení: 100.00 % ◦ Test 'Test nahodnymi daty (Add, Resettle, Print)': Úspěch <ul style="list-style-type: none"> ▪ Dosáženo: 100.00 %, požadováno: 50.00 % ▪ Celková doba běhu: 0.015 s (limit: 3.000 s) ▪ Využití paměti: 12840 KiB (limit: 111017 KiB) ▪ Úspěch v závazném testu, hodnocení: 100.00 % ◦ Test 'Test kopirujiciho konstrukturu': Úspěch <ul style="list-style-type: none"> ▪ Dosáženo: 100.00 %, požadováno: 50.00 % ▪ Celková doba běhu: 0.128 s (limit: 2.985 s) ▪ Využití paměti: 14416 KiB (limit: 111017 KiB) ▪ Úspěch v závazném testu, hodnocení: 100.00 % ◦ Test 'Test operatoru '=': Úspěch <ul style="list-style-type: none"> ▪ Dosáženo: 100.00 %, požadováno: 50.00 % ▪ Celková doba běhu: 0.364 s (limit: 2.857 s) ▪ Využití paměti: 15776 KiB (limit: 111017 KiB) ▪ Úspěch v závazném testu, hodnocení: 100.00 % ◦ Test 'Test nahodnymi hodnotami + test prace s pameti': Úspěch <ul style="list-style-type: none"> ▪ Dosáženo: 100.00 %, požadováno: 50.00 % ▪ Celková doba běhu: 2.720 s (limit: 12.000 s) ▪ Využití paměti: 24156 KiB (limit: 119210 KiB) ▪ Úspěch v závazném testu, hodnocení: 100.00 % ◦ Test 'Efektivita - kopie beze zmen': Úspěch <ul style="list-style-type: none"> ▪ Dosáženo: 100.00 %, požadováno: 70.00 % ▪ Celková doba běhu: 0.009 s (limit: 2.000 s) ▪ Využití paměti: 12836 KiB (limit: 111017 KiB) ▪ Úspěch v nepovinném testu, hodnocení: 100.00 % ◦ Test 'Efektivita - kopie + malo zmen': Úspěch 		

- Dosaženo: 100.00 %, požadováno: 70.00 %
- Celková doba běhu: 1.703 s (limit: 4.000 s)
- Využití paměti: 25008 KiB (limit: 111017 KiB)
- Úspěch v nepovinném testu, hodnocení: 100.00 %
- Test 'Efektivita - test de-duplikace': Úspěch
 - Dosaženo: 100.00 %, požadováno: 100.00 %
 - Celková doba běhu: 1.081 s (limit: 4.000 s)
 - Využití paměti: 22604 KiB (limit: 111017 KiB)
 - Úspěch v bonusovém testu, hodnocení: 120.00 %
- Všechny paměťové bloky byly uvolněné - ok.
- Celkové hodnocení: 120.00 % (= 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.20)
- Použité nápovědy: 1
- Penalizace za vyčerpané nápovědy: Není (1 <= 2 limit)
- Celkové procentní hodnocení: 120.00 %
- Bonus za včasné odevzdání: 1.00
- Celkem bodů: 1.20 * (10.00 + 1.00) = 13.20

		Celkem	Průměr	Maximum	Jméno funkce
	Funkce:	53	--	--	--
SW metriky:	Řádek kódu:	544	10.26 ± 8.81	32	operator =(const CPerson &)
	Cyklomatická složitost:	112	2.11 ± 2.96	12	findString(const char *,int *)

5	31.03.2013 22:04:35	Download
Stav odevzdání:	Ohodnoceno	
Hodnocení:	13.2000	

• **Hodnotitel: automat**

- Program zkompilován
- Test 'Zakladni test s parametry podle ukazky': Úspěch
 - Dosaženo: 100.00 %, požadováno: 100.00 %
 - Celková doba běhu: 0.000 s (limit: 3.000 s)
 - Využití paměti: 12580 KiB (limit: 111017 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test nahodnymi daty (Add, Resettle, Print)': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.016 s (limit: 3.000 s)
 - Využití paměti: 12844 KiB (limit: 111017 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test kopirujiciho konstruktora': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.121 s (limit: 2.984 s)
 - Využití paměti: 14420 KiB (limit: 111017 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test operatoru '=': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.340 s (limit: 2.863 s)
 - Využití paměti: 15792 KiB (limit: 111017 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test nahodnymi hodnotami + test prace s pameti': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 2.715 s (limit: 12.000 s)
 - Využití paměti: 24196 KiB (limit: 119210 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Efektivita - kopie beze zmen': Úspěch
 - Dosaženo: 100.00 %, požadováno: 70.00 %
 - Celková doba běhu: 0.009 s (limit: 2.000 s)
 - Využití paměti: 12840 KiB (limit: 111017 KiB)
 - Úspěch v nepovinném testu, hodnocení: 100.00 %
- Test 'Efektivita - kopie + malo zmen': Úspěch
 - Dosaženo: 100.00 %, požadováno: 70.00 %
 - Celková doba běhu: 1.624 s (limit: 4.000 s)
 - Využití paměti: 25016 KiB (limit: 111017 KiB)
 - Úspěch v nepovinném testu, hodnocení: 100.00 %

- Test 'Efektivita - test de-duplikace': Úspěch
 - Dosaženo: 100.00 %, požadováno: 100.00 %
 - Celková doba běhu: 1.030 s (limit: 4.000 s)
 - Využití paměti: 22616 KiB (limit: 111017 KiB)
 - Úspěch v bonusovém testu, hodnocení: 120.00 %
- Všechny paměťové bloky byly uvolněné - ok.
- Celkové hodnocení: 120.00 % (= 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.20)
- Použité nápovědy: 1
- Penalizace za vyčerpané nápovědy: Není (1 <= 2 limit)
- Celkové procentní hodnocení: 120.00 %
- Bonus za včasné odevzdání: 1.00
- Celkem bodů: 1.20 * (10.00 + 1.00) = 13.20

		Celkem	Průměr	Maximum	Jméno funkce
SW metriky:	Funkce:	54	--	--	--
	Řádek kódu:	627	11.61 ± 13.13	83	main()
	Cyklomatická složitost:	113	2.09 ± 2.93	12	findString(const char *,int *)

4

31.03.2013 18:02:27

[Download](#)

Stav odevzdání: Ohodnoceno

Hodnocení: 7.7000

- **Hodnotitel: automat**
 - Program zkompileován
 - Test 'Zakladni test s parametry podle ukazky': Úspěch
 - Dosaženo: 100.00 %, požadováno: 100.00 %
 - Celková doba běhu: 0.000 s (limit: 3.000 s)
 - Využití paměti: 12576 KiB (limit: 111017 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
 - Test 'Test nahodnymi daty (Add, Resettle, Print)': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.023 s (limit: 3.000 s)
 - Využití paměti: 12840 KiB (limit: 111017 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
 - Test 'Test kopirujiciho konstruktora': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.331 s (limit: 2.977 s)
 - Využití paměti: 14408 KiB (limit: 111017 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
 - Test 'Test operatoru '=': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 1.137 s (limit: 2.646 s)
 - Využití paměti: 15844 KiB (limit: 111017 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
 - Test 'Test nahodnymi hodnotami + test prace s pameti': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 2.724 s (limit: 12.000 s)
 - Využití paměti: 24196 KiB (limit: 119210 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
 - Test 'Efektivita - kopie beze zmen': Úspěch
 - Dosaženo: 100.00 %, požadováno: 70.00 %
 - Celková doba běhu: 0.015 s (limit: 2.000 s)
 - Využití paměti: 12836 KiB (limit: 111017 KiB)
 - Úspěch v nepovinném testu, hodnocení: 100.00 %
 - Test 'Efektivita - kopie + malo zmen': Program překročil přidělenou maximální dobu běhu
 - Vyčerpání limitu na celý test, program násilně ukončen po: 4.005 s (limit: 4.000 s)
 - Neúspěch v nepovinném testu, hodnocení: 70.00 %
 - Celkové hodnocení: 70.00 % (= 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 0.70)
- Použité nápovědy: 1
- Penalizace za vyčerpané nápovědy: Není (1 <= 2 limit)
- Celkové procentní hodnocení: 70.00 %
- Bonus za včasné odevzdání: 1.00
- Celkem bodů: 0.70 * (10.00 + 1.00) = 7.70

		Celkem	Průměr	Maximum	Jméno funkce
	Funkce:	51	--	-- --	
SW metrikky:	Řádek kódu:	623	12.22 ± 13.68	84	main()
	Cyklomatická složitost:	111	2.18 ± 2.99	12	findPerson(const char *,int *)

3

31.03.2013 16:11:01

Download

Stav odevzdání:

Ohodnoceno

Hodnocení:

7.7000

Hodnotitel: automat

Program zkompilován

Test 'Zakladni test s parametry podle ukazky': Úspěch

Dosaženo: 100.00 %, požadováno: 100.00 %

Celková doba běhu: 0.000 s (limit: 3.000 s)

Využití paměti: 12576 KiB (limit: 111017 KiB)

Úspěch v závazném testu, hodnocení: 100.00 %

Test 'Test nahodnymi daty (Add, Resettle, Print)': Úspěch

Dosaženo: 100.00 %, požadováno: 50.00 %

Celková doba běhu: 0.028 s (limit: 3.000 s)

Využití paměti: 12840 KiB (limit: 111017 KiB)

Úspěch v závazném testu, hodnocení: 100.00 %

Test 'Test kopirujiciho konstrukturu': Úspěch

Dosaženo: 100.00 %, požadováno: 50.00 %

Celková doba běhu: 0.356 s (limit: 2.972 s)

Využití paměti: 14436 KiB (limit: 111017 KiB)

Úspěch v závazném testu, hodnocení: 100.00 %

Test 'Test operatoru '=': Úspěch

Dosaženo: 100.00 %, požadováno: 50.00 %

Celková doba běhu: 1.135 s (limit: 2.616 s)

Využití paměti: 15736 KiB (limit: 111017 KiB)

Úspěch v závazném testu, hodnocení: 100.00 %

Test 'Test nahodnymi hodnotami + test prace s pameti': Úspěch

Dosaženo: 100.00 %, požadováno: 50.00 %

Celková doba běhu: 2.817 s (limit: 12.000 s)

Využití paměti: 24212 KiB (limit: 119210 KiB)

Úspěch v závazném testu, hodnocení: 100.00 %

Test 'Efektivita - kopie beze zmen': Úspěch

Dosaženo: 100.00 %, požadováno: 70.00 %

Celková doba běhu: 0.015 s (limit: 2.000 s)

Využití paměti: 12836 KiB (limit: 111017 KiB)

Úspěch v nepovinném testu, hodnocení: 100.00 %

Test 'Efektivita - kopie + malo zmen': Program překročil přidělenou maximální dobu běhu

Vyčerpání limitu na celý test, program násilně ukončen po: 4.004 s (limit: 4.000 s)

Neúspěch v nepovinném testu, hodnocení: 70.00 %

Celkové hodnocení: 70.00 % (= 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 0.70)

Použité nápovědy: 1

Penalizace za vyčerpané nápovědy: Není (1 <= 2 limit)

Celkové procentní hodnocení: 70.00 %

Bonus za včasné odevzdání: 1.00

Celkem bodů: 0.70 * (10.00 + 1.00) = 7.70

SW metriky:

Funkce:

58

Řádek kódu:

653

Cyklomatická složitost:

118

Průměr

11.26 ± 13.00

2.03 ± 2.83

Maximum

84

12

Jméno funkce

main()

findPerson(const char *,int *)

2	31.03.2013 15:37:45	Download			
Stav odevzdání:		Ohodnoceno			

Hodnocení: 7.7000

• Hodnotitel: automat

- Program zkompileován
- Test 'Zakladni test s parametry podle ukazky': Úspěch
 - Dosaženo: 100.00 %, požadováno: 100.00 %
 - Celková doba běhu: 0.000 s (limit: 3.000 s)
 - Využití paměti: 12576 KiB (limit: 111017 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test nahodnymi daty (Add, Resettle, Print)': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.026 s (limit: 3.000 s)
 - Využití paměti: 12824 KiB (limit: 111017 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test kopirujícího konstruktora': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.386 s (limit: 2.974 s)
 - Využití paměti: 14544 KiB (limit: 111017 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test operatoru '=': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 1.351 s (limit: 2.588 s)
 - Využití paměti: 15912 KiB (limit: 111017 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test nahodnymi hodnotami + test prace s pameti': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 3.225 s (limit: 12.000 s)
 - Využití paměti: 24272 KiB (limit: 119210 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Efektivita - kopie beze zmen': Úspěch
 - Dosaženo: 100.00 %, požadováno: 70.00 %
 - Celková doba běhu: 0.018 s (limit: 2.000 s)
 - Využití paměti: 12832 KiB (limit: 111017 KiB)
 - Úspěch v nepovinném testu, hodnocení: 100.00 %
- Test 'Efektivita - kopie + malo zmen': Program překročil přidělenou maximální dobu běhu
 - Vyčerpání limitu na celý test, program násilně ukončen po: 4.004 s (limit: 4.000 s)
 - Neúspěch v nepovinném testu, hodnocení: 70.00 %
- Celkové hodnocení: 70.00 % (= 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 0.70)
- Použité nápovědy: 1
- Penalizace za vyčerpané nápovědy: Není (1 <= 2 limit)
- Celkové procentní hodnocení: 70.00 %
- Bonus za včasné odevzdání: 1.00
- Celkem bodů: 0.70 * (10.00 + 1.00) = 7.70

		Celkem	Průměr	Maximum	Jméno funkce
SW metriky:	Funkce:	57	--	--	--
	Řádek kódu:	639	11.21 ± 13.09	84	main()
	Cyklomatická složitost:	115	2.02 ± 2.86	12	findPerson(const char *,int *)

1 31.03.2013 13:54:04 Download

Stav odevzdání: Ohodnoceno

Hodnocení: 6.9300

• Hodnotitel: automat

- Chyba při kompilaci v režimu 'pedantic' - 10% penalizace [Zpřístupnit nápovědu (198 B)]
- Test 'Zakladni test s parametry podle ukazky': Úspěch
 - Dosaženo: 100.00 %, požadováno: 100.00 %
 - Celková doba běhu: 0.000 s (limit: 3.000 s)
 - Využití paměti: 12576 KiB (limit: 111017 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test nahodnymi daty (Add, Resettle, Print)': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.012 s (limit: 3.000 s)

- Využití paměti: 13104 KiB (limit: 111017 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test kopírujícího konstruktora': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.081 s (limit: 2.988 s)
 - Využití paměti: 16128 KiB (limit: 111017 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test operatoru '=': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 0.214 s (limit: 2.907 s)
 - Využití paměti: 18444 KiB (limit: 111017 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test nahodnými hodnotami + test práce s pamětí': Úspěch
 - Dosaženo: 100.00 %, požadováno: 50.00 %
 - Celková doba běhu: 3.131 s (limit: 12.000 s)
 - Využití paměti: 26316 KiB (limit: 119210 KiB)
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Efektivita - kopie beze změn': Úspěch
 - Dosaženo: 100.00 %, požadováno: 70.00 %
 - Celková doba běhu: 0.008 s (limit: 2.000 s)
 - Využití paměti: 14684 KiB (limit: 111017 KiB)
 - Úspěch v nepovinném testu, hodnocení: 100.00 %
- Test 'Efektivita - kopie + malo změn': Neúspěch
 - Dosaženo: 0.00 %, požadováno: 70.00 %
 - Celková doba běhu: 0.734 s (limit: 4.000 s)
 - Využití paměti: 60800 KiB (limit: 111017 KiB)
 - Neúspěch v nepovinném testu, hodnocení: 70.00 %
 - ☐ Nesprávný výstup
- Test 'Efektivita - test de-duplikace': Překročen paměťový limit
 - Celková doba běhu: 0.997 s (limit: 4.000 s)
 - Využití paměti: 231164 KiB (limit: 111017 KiB)
 - Neúspěch v bonusovém testu, hodnocení: Bonus nebude udělen
 - Nesprávný výstup **[Zpřístupnit nápovědu (71 B)]**
- Všechny paměťové bloky byly uvolněné - ok.
- Celkové hodnocení: 63.00 % (= (1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 0.70) * 0.9)
- Celkové procentní hodnocení: 63.00 %
- Bonus za včasné odevzdání: 1.00
- Celkem bodů: 0.63 * (10.00 + 1.00) = 6.93

		Celkem	Průměr	Maximum	Jméno funkce
SW metriky:	Funkce:	50	--	-- --	
	Řádek kódu:	500	10.00 ± 12.64	82	main()
	Cyklomatická složitost:	92	1.84 ± 2.64	12	findPerson(const char *,int *)