**РУСЕНСКИ УНИВЕРСИТЕТ “АНГЕЛ КЪНЧЕВ”**

Катедра: **Компютърни системи и технологии**

КУРСОВА РАБОТА  
**ПО ДИСКРЕТНИ СТРУКТУРИ И МОДЕЛИРАНЕ**

Студент:

Фак. номер:

Група:

Специалност:

Дата: Изготвил:  
 Проверил:

/ *доц. д-р И. И. Желязкова*/

**Съдържание**

[1 Задание 3](#_Toc347871329)

[2 Архитектура на средата 4](#_Toc347871330)

[3 Описание на използваните функции 5](#_Toc347871331)

[4 Инструкции за работа с програмата 7](#_Toc347871332)

[5 Среда за програмиране 14](#_Toc347871333)

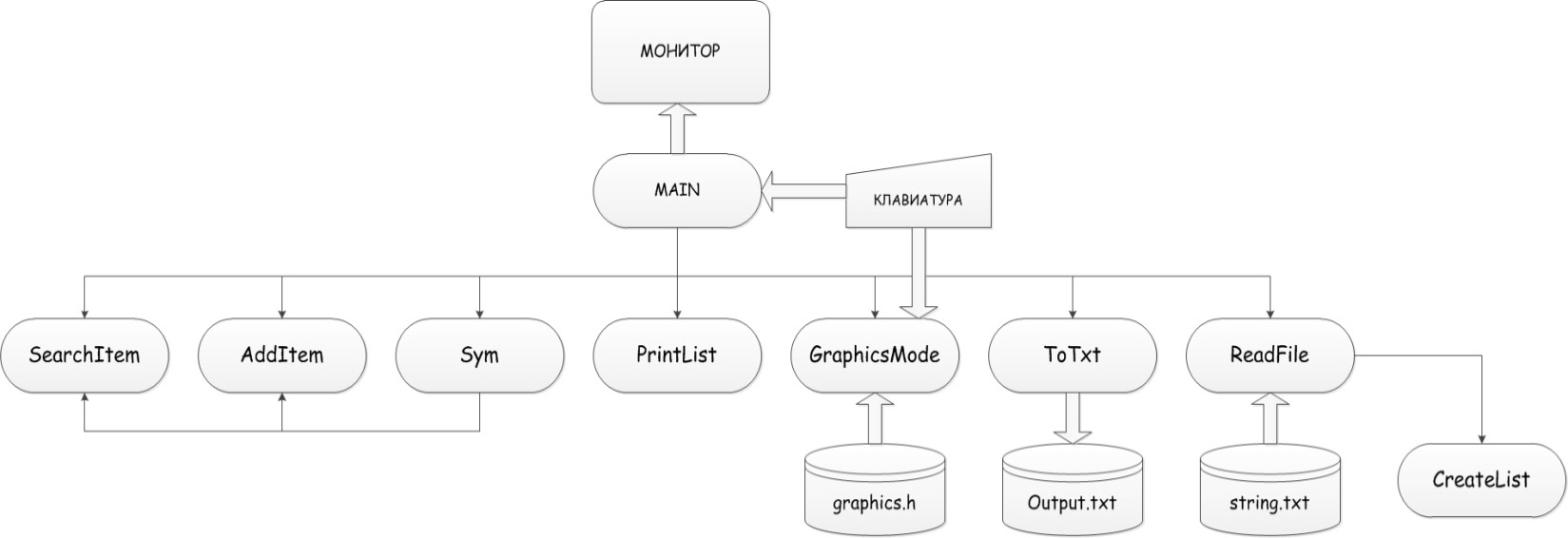
[6 Код на програмата 15](#_Toc347871334)

# Задание

Да се проектира и реализира инструментално средство за моделиране с множества, работещо в цветен текстов и графичен режим на монитора и реализиращо следните изисквания:

* въвеждане на стрингови множества от текстов файл;
* изпълнение и графично представяне на операцията допълнение на множество;
* изпълнение и графично представяне на операцията симетрична разлика на две множества;
* проверка за принадлежност на елемент към множество;
* записване на данните и резултатите от втората операция в текстов файл.

# Архитектура на средата



# Описание на използваните функции

1. Вид на функцията: **int main();**

Действие: **главна функция;**  
 Параметри: **няма;**  
 Използвани глобални променливи: **HANDLE hConsole, Point HeadA, HeadB, HeadC;**  
 Извиквани функции:**ReadFile(), PrintList(), ToTxt(), SearchItem(), Sym(), AddItem(), GraphicsMode();**

1. Вид на функцията: **void ReadFile( char FileName [] );**

Действие: **Функция за въвеждане на стрингови множества от текстов файл;**  
 Параметри: **char FileName [] –** име на текстовия файл за четене.  
 Използвани глобални променливи: **int i, Point HeadA, HeadB, ifstream txt;**  
 Извиквани функции: **CreateList();**

1. Вид на функцията: **CreateList( string buf, Point &Head );**

Действие: **Функция за създаване на списък (множество);**  
 Параметри: **string buf –** стрингът за записване в списъка; **Point &Head –** указател към първия елемент на списъка**;**  
 Използвани глобални променливи: **няма;**  
 Извиквани функции: **няма;**

1. Вид на функцията: **void AddItem( string buf, Point Head );**

Действие: **Функция за добавяне на нов елемент към списък (множество);**  
 Параметри: **string buf –** стрингът за записване в списъка; **Point Head –** указател към първия елемент на списъка**;**  
 Използвани глобални променливи: **няма;**  
 Извиквани функции: **няма;**

1. Вид на функцията: **void PrintList( Point Head );**

Действие: **Функция за извеждане на списък (множество) на екрана;**  
 Параметри: **Point Head –** указател към първия елемент на списъка**;**  
 Използвани глобални променливи: **няма;**  
 Извиквани функции: **няма;**

1. Вид на функцията: **bool SearchItem( string buf, Point Head );**

Действие:**Функция за търсене на елемент в даден списък;**  
 Параметри: **string buf –** стрингът за записване в списъка; **Point Head –** указател към първия елемент на списъка**;**

Използвани глобални променливи: **няма;**  
 Извиквани функции: **няма;**

1. Вид на функцията: **void Sym( Point HeadA, Point HeadB );**

Действие: **Функция за намиране симетричната разлика на две множества;**  
 Параметри: **Point HeadА, Point HeadB –** указатели към първия елемент на съответните списъци**;**

Използвани глобални променливи: **Point HeadC –** указател към първия елемент на резултантния списък (множество);  
 Извиквани функции: **SearchItem(), AddItem().**

1. Вид на функцията: **void ToTxt( Point Head );**

Действие: **Функция за записване на данните от втората операция (допълване) в текстов файл;**  
 Параметри: **Point Head –** указател към първия елемент на списъка**;**

Използвани глобални променливи: **ofstream OutputTxt;**

Извиквани функции: **няма;**

1. Вид на функцията: **void GraphicMode( Point Head, int Operation );**

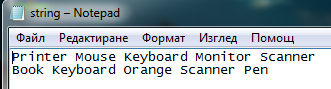
Действие:**Функция за графично представяне на операциите допълнение и симетрична разлика ;**  
 Параметри **Point Head –** указател към първия елемент на списъка**;**

**Int Operation** – код на операцията – допълнение/симетрична разлика;

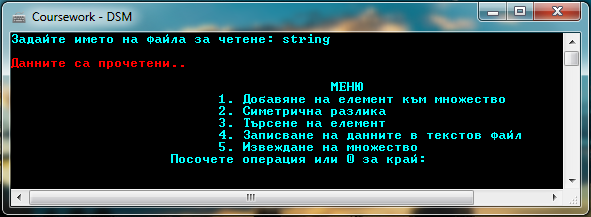
Използвани глобални променливи: **int i, gd, gm;**  
 Извиквани функции: **няма.**

# Инструкции за работа с програмата

Със стартиране на програмния файл \*.exe се изисква въвеждане на името на текстовия файл (Фиг. 1) за четене. Не е необходимо въвеждането на разширението (.txt). Ако файлът е намерен се появява меню, от коeто се избират съответните операции (Фиг. 2). В случай, че не е намерен, след потвърждение, се затваря прорамата.



Фиг. Текстов файл - string.txt



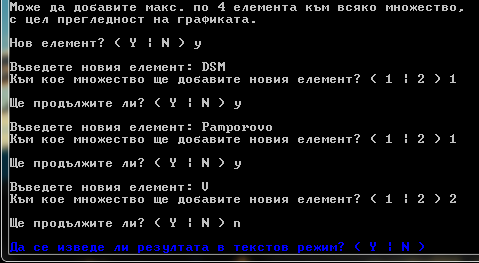
Фиг. Избиране на текстовия файл

При избор на **операция 1 (Добавяне на елемент към множество)** от менюто се появява следното съобщение (Фиг. 3).



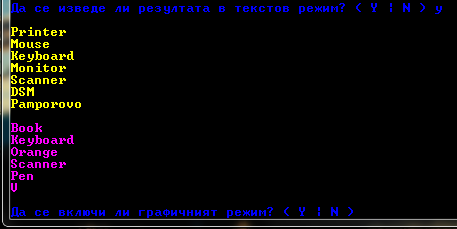
Фиг. Съобщение

Това е така, защото е предвидено макс. по 5 елемента за всяко множество, които се добавят от файла, избран в началото на програмата и може да се добавят още макс. по 4 елемента с цел прегледност на графиката. След въвеждане на “Y” се въвежда стринг (дума), след което се избира множеството (1 или 2), към което да се добави новият елемент. По желание може да се добавят още елементи (Фиг. 4).

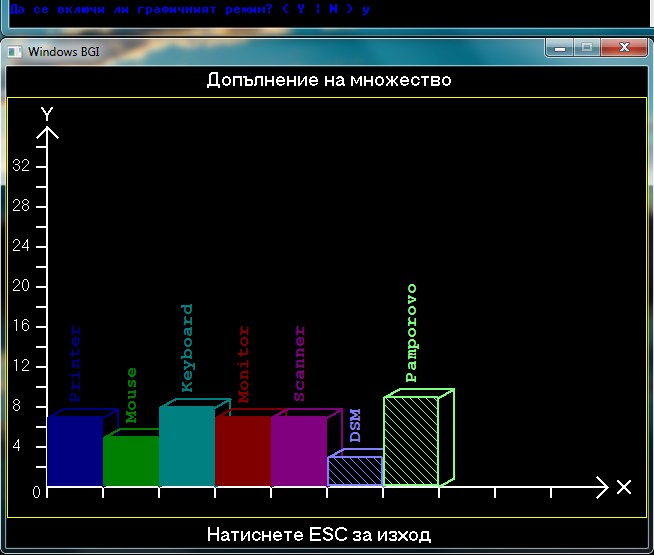


Фиг. Добавяне на елемент към множество

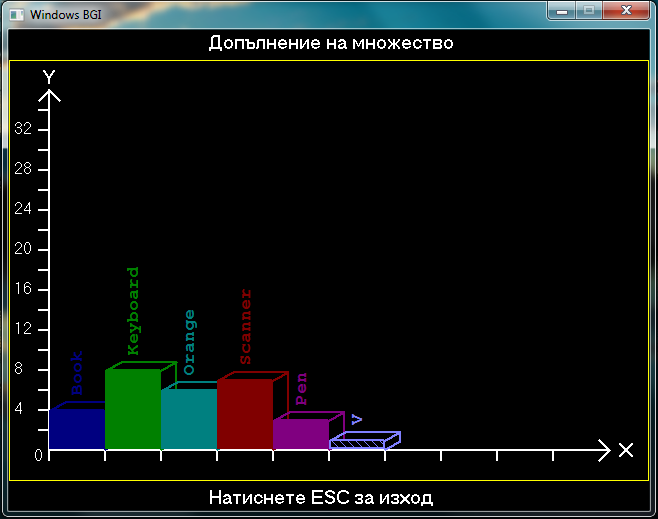
След добавянето, програмата дава възможност за визуализиране на резултата както в текстов (Фиг. 5), така и в графичен режим (Фиг. 6 и Фиг. 7).Като първо се извежда първото множество (с жълт цвят на текста) - Фиг. 6, а след това второто (магента) - Фиг. 7.



Фиг. Допълнение на множество (текстов режим)

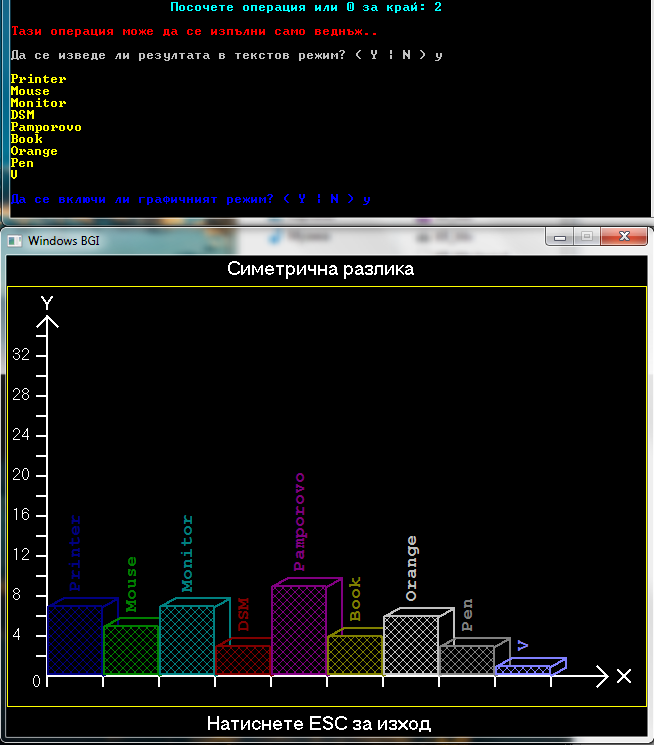


Фиг. Допълнение на първото множество (графичен режим)



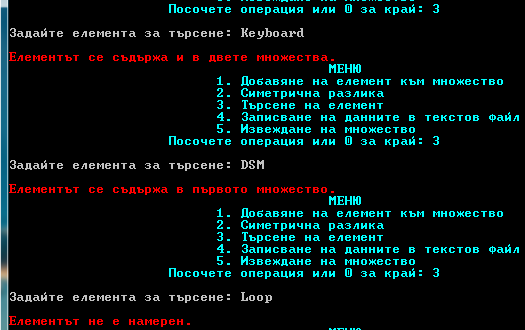
Фиг. Допълнение на второто множество (графичен режим)

**Операция 2 (Симетрична разлика)** – тази операция може да се изпълни еднократно. След натискане на произволен клавиш, се избира дали да се визуализира крайният резултат (в текстов и графичен режим) (Фиг. 8).



Фиг. Симетрична разлика

**Операция 3 (Търсене на елемент)** – 3адава се елементът за търсене. Зададеният елемент се търси в двете множества и резултатът се извежда на екрана (Фиг. 9).

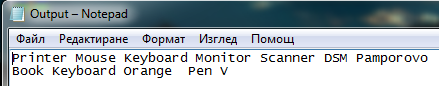


Фиг. Търсене на елемент

**Операция 4 (Записване на данните в текстов файл)** – ако е избрана операция 1 (Фиг. 10) данните се записват в текстовия файл “Output.txt“ (Фиг. 11), който се създава в директорията, където е програмата. Ако файлът вече е създаден данните се добавят в края на файла, т.е съдържанието му се запазва.

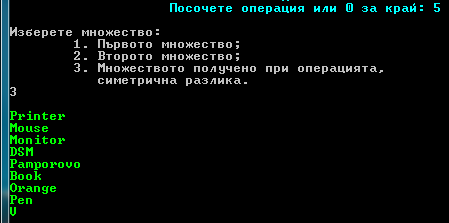


Фиг. Записване на данните в текстов файл



Фиг. Output.txt

**Операция 5 (Извеждане на множество)** – тази операция служи за извеждане на дадено множество на екрана в текстов режим. Вариантите са 3, съответно първото, второто и третото множество. Последното се появява сред вариантите след като е изпълнена операция 2, иначе се „скрива“ (Фиг. 12).



Фиг. Извеждане на множество

За изход от програмата е необходимо да се въведе нула - „0“ (Фиг. 13).



Фиг. Изход от програмата

# Среда за програмиране

Поради спецификата на заданието на курсовата работа изборът на среда за програмиране е сведен до избор на Dev C++ 4.9.9.2. С помощта на модула (WinBGIm), който се вгражда допълнително, програмистът има достъп до графични функции.

# Код на програмата

#include <iostream>

#include <fstream>

#include <graphics.h>

#define COUNT 5

#define SYM 1

#define ADD 2

#define ESC 0x1b

**using** **namespace** std;

**typedef** **struct** Item {

string str;

Item \*next;

}\*Point;

Point HeadA, HeadB, HeadC;

ifstream txt;

ofstream OutputTxt;

**int** i;

**int** gd = DETECT, gm;

HANDLE hConsole;

*//Създаване на списък - тип опашка (FIFO)*

**void** CreateList( string buf, Point &Head ) {

Point P, Last;

P = **new** Item;

P->str = buf;

P->next = NULL;

**if** ( !Head )

Head = P;

**else**

Last->next = P;

Last = P;

}

*//Въвеждане на стрингови множества от текстов файл*

**void** ReadFile( **char** FileName [] ) {

string buffer;

**if** ( txt ) {

i = 0;

txt.open( FileName, ios::in );

**while** ( !txt.eof() && txt >> buffer && i < COUNT) {

CreateList( buffer, HeadA );

i++;

}*//while*

i = 0;

**int** length = buffer.length();

txt.seekg (-length, ios::cur);

**while** ( !txt.eof() && txt >> buffer && i < COUNT) {

CreateList( buffer, HeadB );

i++;

}*//while*

txt.close();

}*//if*

**if** ( !txt ) {

cout << endl << "Файлът не е намерен!" << endl;

cout << "\nНатиснете произволен клавиш за край..";

getchar();

exit(1);

}

}

*//Вмъкване в края на списъка..*

**void** AddItem( string buf, Point Head ) {

Point P = Head, Q;

Q = **new** Item;

Q->str = buf;

Q->next = NULL;

**while** ( P->next )

P = P->next;

P->next = Q;

}

*//Извеждане*

**void** PrintList( Point Head ) {

Point H = Head;

**while** ( H ) {

cout << endl << H->str;

H = H->next;

}

cout << endl;

}

*//Търсене на елемент*

**bool** SearchItem( string buf, Point Head ) {

**while** ( Head && buf != Head->str )

Head = Head->next;

**if** (Head){

**if** ( buf == Head->str )

**return** **true**;

}

**else**

**return** **false**;

}

*//Симетрична разлика*

**void** Sym( Point HeadA, Point HeadB ) {

Point hA = HeadA;

**while** (HeadA){

**if** ( !SearchItem(HeadA->str, HeadB) )

AddItem(HeadA->str, HeadC);

HeadA = HeadA->next;

}

**while** (HeadB){

**if** ( !SearchItem(HeadB->str, hA) )

AddItem(HeadB->str, HeadC);

HeadB = HeadB->next;

}

}

*//Записване на данните от втората операция в текстов файл*

**void** ToTxt( Point Head ) {

Point H = Head;

OutputTxt.open( "Output.txt", ios::app );

**while** ( H ) {

OutputTxt << H->str << " ";

H = H->next;

}

OutputTxt << endl;

OutputTxt.close();

}

*/\*///////////////Graphics/////////////////\*/*

**void** GraphicMode( Point Head, **int** Operation ) {

**int** next\_col = 1; *//next color*

**char** buffer[3];

**char** \*msg;

**int** j;

initgraph( &gd, &gm, "C:\\TC\\BGI" );

*/\*Design\*/*

setcolor( YELLOW );

rectangle( 0, 30, 639, 450 );

settextstyle( SANS\_SERIF\_FONT, HORIZ\_DIR, 2 );

setcolor( WHITE );

**if** ( Operation == ADD )

outtextxy( 200, 0, "Допълнение на множество" );

**else** **if** ( Operation == SYM )

outtextxy( 220, 0, "Симетрична разлика" );

outtextxy( 200, 455, "Натиснете ESC за изход" );

setlinestyle( SOLID\_LINE, 0, 2 );

line( 40, 420, 40, 60 );

line( 40, 420, 600, 420 );

line( 30, 70, 40, 60 );

line( 50, 70, 40, 60 );

line( 590, 410, 600, 420 );

line( 590, 430, 600, 420 );

**for**( i = 400; i > 70; i -= 20 )

line(30, i, 40, i);

**for**( i = 40; i < 600; i += 56 )

line( i, 420, i, 430 );

outtextxy( 34, 35, "Y" );

outtextxy( 610, 408, "X" );

settextstyle( SANS\_SERIF\_FONT, HORIZ\_DIR, 1 );

outtextxy( 25, 415, "0");

**for** ( i = 4, j = 0; i < 36; i += 4, j += 40 ){

itoa( i, buffer, 10 );

outtextxy( 5, 368-j, buffer );

}

*/\*End Design\*/*

i = 0;

j = 0;

settextstyle( 10, VERT\_DIR, 1 );

**while**( Head && j < 9) {

**if** ( Operation == ADD ) {

**if** ( j < COUNT ) {

setfillstyle( SOLID\_FILL, next\_col );

setcolor( next\_col );

}

**else** {

setfillstyle( LTBKSLASH\_FILL, next\_col + 3 );

setcolor( next\_col + 3 );

}

}

**else** **if** ( Operation == SYM ) {

setfillstyle( XHATCH\_FILL, next\_col );

setcolor( next\_col );

}

bar3d( 41+i, 420-( Head->str.length()\*10 ), 96+i, 419, 15, 1 );

msg = **new** **char**[ Head->str.length() ];

strcpy(msg, Head->str.c\_str());

outtextxy( 58+i, 405-( Head->str.length()\*10 ), msg );

Head = Head->next;

next\_col++;

i += 56;

j++;

}

*/\*Press ESC to terminate\*/*

PressESC:

**while**( !kbhit() );

**if** ( getch() == ESC )

closegraph();

**else** **goto** PressESC;

*/\*End terminate\*/*

}

*/\*//////////////End Graphics//////////////\*/*

**int** main() {

setlocale( LC\_ALL, "bgr\_BGR" );*//Kирилица*

SetConsoleTitle("Coursework - DSM");

**char** FileName[26], ch, chGr;

**short** **int** chList, cA = 0, cB = 0, op;

string buf;

**bool** result[1], flag = **true**;

hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

HeadA = NULL;

HeadB = NULL;

HeadC = **new** Item;

HeadC->next = NULL;

SetConsoleTextAttribute(hConsole, LIGHTCYAN);

printf( "Задайте името на файла за четене: " );

gets( FileName );

strcat( FileName, ".txt" );

ReadFile ( FileName );

SetConsoleTextAttribute(hConsole, LIGHTRED);

cout << "\nДанните са прочетени..\n";

**do** {

SetConsoleTextAttribute(hConsole, LIGHTCYAN);

printf( "\n%40sМЕНЮ\n","" );

printf( "%26s1. Добавяне на елемент към множество\n","" );

printf( "%26s2. Симетрична разлика\n","" );

printf( "%26s3. Търсене на елемент\n","" );

printf( "%26s4. Записване на данните в текстов файл\n","" );

printf( "%26s5. Извеждане на множество\n","" );

printf( "%20sПосочете операция или 0 за край: ","" );

scanf( "%d", &op ); getchar();

**switch** ( op ) {

*/\*//////CASE 1//////\*/*

**case** 1:

SetConsoleTextAttribute(hConsole, LIGHTGRAY);

cout << endl << "Може да добавите макс. по 4 елемента към всякo множество,"

<< "\nс цел прегледност на графиката.\n";

cout << endl << "Нов елемент? ( Y | N ) ";

cin >> ch;

**while** ( ch != 'n' && ch != 'N' && ch != 'н' && ch != 'Н') {

cout << endl << "Въведете новия елемент: ";

cin >> buf;

again1:

cout << "Към кое множество ще добавите новия елемент? ( 1 | 2 ) ";

cin >> chList;

**if** ( chList == 1 && cA < 4 ) {

AddItem( buf, HeadA );

cA++;

}

**else** **if** ( chList == 2 && cB < 4) {

AddItem( buf, HeadB );

cB++;

}

**else** {

SetConsoleTextAttribute(hConsole, LIGHTRED);

cout << endl << "Неправилен избор!" << endl;

**goto** again1;

}*//else*

SetConsoleTextAttribute(hConsole, LIGHTGRAY);

cout << endl << "Ще продължите ли? ( Y | N ) ";

cin >> ch;

}*//while*

SetConsoleTextAttribute(hConsole, LIGHTBLUE);

cout << endl <<"Да се изведе ли резултата в текстов режим? ( Y | N ) ";

cin >> chGr;

**if** ( chGr == 'y' || chGr == 'Y' ) {

SetConsoleTextAttribute(hConsole, YELLOW);

PrintList( HeadA );

SetConsoleTextAttribute(hConsole, LIGHTMAGENTA);

PrintList( HeadB );

}

SetConsoleTextAttribute(hConsole, LIGHTBLUE);

cout << endl <<"Да се включи ли графичният режим? ( Y | N ) ";

cin >> chGr;

**if** ( chGr == 'y' || chGr == 'Y' ) {

GraphicMode( HeadA, ADD );

GraphicMode( HeadB, ADD );

}

**break**;

*/\*//////END CASE 1//////\*/*

*/\*//////CASE 2//////\*/*

**case** 2:

SetConsoleTextAttribute(hConsole, LIGHTRED);

cout << endl <<"Тази операция може да се изпълни само веднъж.. ";

getchar();

**if** ( flag ) {

flag = **false**;

Sym( HeadA, HeadB );

SetConsoleTextAttribute(hConsole, LIGHTGRAY);

cout << endl <<"Да се изведе ли резултата в текстов режим? ( Y | N ) ";

cin >> chGr;

**if** ( chGr == 'y' || chGr == 'Y' ) {

SetConsoleTextAttribute(hConsole, YELLOW);

PrintList( HeadC->next );

}

SetConsoleTextAttribute(hConsole, LIGHTBLUE);

cout << endl <<"Да се включи ли графичният режим? ( Y | N ) ";

cin >> chGr;

**if** ( chGr == 'y' || chGr == 'Y' )

GraphicMode( HeadC->next, SYM );

}

**break**;

*/\*//////END CASE 2//////\*/*

*/\*//////CASE 3//////\*/*

**case** 3:

SetConsoleTextAttribute(hConsole, LIGHTGRAY);

cout << endl << "Задайте елемента за търсене: ";

cin >> buf;

result[0] = SearchItem(buf, HeadA);

result[1] = SearchItem(buf, HeadB);

SetConsoleTextAttribute(hConsole, LIGHTRED);

**if** ( result[0] && result[1] )

cout << endl << "Елементът се съдържа и в двете множества.";

**else** **if** ( result[0] )

cout << endl << "Елементът се съдържа в първото множество.";

**else** **if** ( result[1] )

cout << endl << "Елементът се съдържа във второто множество.";

**else**

cout << endl << "Елементът не е намерен.";

**break**;

*/\*//////END CASE 3//////\*/*

*/\*//////CASE 4//////\*/*

**case** 4:

ToTxt( HeadA );

ToTxt( HeadB );

SetConsoleTextAttribute(hConsole, LIGHTRED);

cout << "\nДанните са записани в \"Output.txt\"..";

getchar();

**break**;

*/\*//////END CASE 4//////\*/*

*/\*//////CASE 5//////\*/*

**case** 5:

SetConsoleTextAttribute(hConsole, LIGHTGRAY);

again2:

cout << "\nИзберете множество: ";

cout << "\n\t1. Първото множество;";

cout << "\n\t2. Второто множество;";

**if** ( HeadC->next )

cout << "\n\t3. Множеството получено при операцията,"

<< "\n\t симетрична разлика.";

cout << endl;

cin >> chList;

**if** ( chList == 1 ) {

SetConsoleTextAttribute(hConsole, YELLOW);

PrintList( HeadA );

}

**else** **if** ( chList == 2 ) {

SetConsoleTextAttribute(hConsole, LIGHTMAGENTA);

PrintList( HeadB );

}

**else** **if** ( HeadC->next && chList == 3 ) {

SetConsoleTextAttribute(hConsole, LIGHTGREEN);

PrintList( HeadC->next );

}

**else** {

SetConsoleTextAttribute(hConsole, LIGHTGRAY);

cout << endl << "Неправилен избор!" << endl;

**goto** again2;

}

**break**;

*/\*//////END CASE 5//////\*/*

}*//switch*

} **while** ( op != 0 );

cout << endl;

cout << "\nНатиснете произволен клавиш за край..";

getchar();

**return**( 0 );

}