Chapter 3.

Lab. 3.2.12 (1)

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

int main(void)

{

int maxball;

int ballsno;

cout << "Max ball number? ";

cin >> maxball;

cout << "How many balls? ";

cin >> ballsno;

if (ballsno > maxball)

{

cout << "Balls number should be less than max ball number!" << endl;

return 0;

}

srand(time(NULL));

int\* balls = new int[ballsno];

for (int i = 0; i < ballsno; ++i)

{

bool newBall;

do

{

balls[i] = (rand() % maxball) + 1;

newBall = true;

for (int j = 0; j < i; ++j)

if (balls[j] == balls[i])

newBall = false;

} while (!newBall);

}

for (int i = 0; i < ballsno; ++i)

cout << balls[i] << ' ';

cout << endl;

delete[] balls;

return 0;

}

3.2.12 (2)

#include <iostream>

#include <ctime>

#include <string>

using namespace std;

struct Collection

{

int elno;

int\* elements;

};

void AddToCollection(Collection& col, int element)

{

if (col.elno == 0)

{

col.elements = new int[1];

col.elements[0] = element;

col.elno += 1;

}

else

{

int\* vector = new int[col.elno + 1];

for (int i = 0; i < col.elno; i++)

{

vector[i] = col.elements[i];

}

vector[col.elno] = element;

delete[] col.elements;

col.elements = vector;

col.elno += 1;

}

}

void PrintCollection(Collection col)

{

cout << "[ ";

for (int i = 0; i < col.elno; i++)

cout << col.elements[i] << " ";

cout << "]" << endl;

}

int main(void)

{

Collection collection = { 0, NULL };

int elems;

cout << "How many elements? ";

cin >> elems;

srand(time(NULL));

for (int i = 0; i < elems; i++)

AddToCollection(collection, elems % 100 + 1);

PrintCollection(collection);

delete[] collection.elements;

return 0;

}

3.4.7 (1)

#include <iostream>

using namespace std;

int main()

{

int vector[] = { 3, -5, 7, 10, -4, 14, 5, 2, -13 };

int n = sizeof(vector) / sizeof(vector[0]);

int res = \*(vector);

for (int i = 0; i < n; i++)

{

if (res > \* (vector + i))

res = \*(vector + i);

}

cout << res;

return 0;

}

3.4.7 (2) #include <iostream>

using namespace std;

int main()

{

int matrix[10][10] = { };

for (int i = 0; i < 10; i++)

{

for (int j = 0; j < 10; j++)

{

\*(\*(matrix + i) + j) = (i + 1) \* (j + 1);

}

}

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

cout.width(4);

cout << matrix[i][j];

}

cout << endl;

}

return 0;

}

3.4.7 (3)

#include <iostream>

#include <ctime>

using namespace std;

bool isLeap(int year)

{

if (year % 4 != 0)

{

return false;

}

else if (year % 100 != 0)

{

return true;

}

else if (year % 400 != 0)

{

return false;

}

else

{

return true;

}

}

int main()

{

for (int yr = 1995; yr < 2017; yr++)

cout << yr << " -> " << isLeap(yr) << endl;

return 0;

}

3.4.7 (4)

#include <iostream>

#include <ctime>

using namespace std;

bool isLeap(int year)

{

if (year % 4 != 0)

{

return false;

}

else if (year % 100 != 0)

{

return true;

}

else if (year % 400 != 0)

{

return false;

}

else

{

return true;

}

}

int monthLength(int year, int month)

{

if (month == 2)

{

if ((year % 400 == 0) || (year % 4 == 0 && year % 100 != 0))

return 29;

else

return 28;

}

else if (month == 1 || month == 3 || month == 5 || month == 7 || month == 8

|| month == 10 || month == 12)

return 31;

else

return 30;

}

int main()

{

for (int yr = 2000; yr < 2002; yr++)

{

for (int mo = 1; mo <= 12; mo++)

cout << monthLength(yr, mo) << " ";

cout << endl;

}

return 0;

}

3.4.7 (5) #include <iostream>

#include <ctime>

using namespace std;

bool isLeap(int year)

{

if (year % 4 != 0)

{

return false;

}

else if (year % 100 != 0)

{

return true;

}

else if (year % 400 != 0)

{

return false;

}

else

{

return true;

}

}

int monthLength(int year, int month)

{

if (month == 2)

{

if ((year % 400 == 0) || (year % 4 == 0 && year % 100 != 0))

return 29;

else

return 28;

}

else if (month == 1 || month == 3 || month == 5 || month == 7 || month == 8

|| month == 10 || month == 12)

return 31;

else

return 30;

}

struct Date

{

int year;

int month;

int day;

};

int dayOfYear(Date date)

{

int daysNumb = 0;

for (int i = 1; i < date.month; i++)

{

daysNumb += monthLength(date.year, i);

}

return daysNumb + date.day;

}

int main()

{

Date d;

cout << "Enter year month day: ";

cin >> d.year >> d.month >> d.day;

cout << dayOfYear(d) << endl;

return 0;

}

3.4.7 (6)

#include <iostream>

#include <ctime>

using namespace std;

bool isLeap(int year)

{

if (year % 4 != 0)

{

return false;

}

else if (year % 100 != 0)

{

return true;

}

else if (year % 400 != 0)

{

return false;

}

else

{

return true;

}

}

int monthLength(int year, int month)

{

if (month == 2)

{

if ((year % 400 == 0) || (year % 4 == 0 && year % 100 != 0))

return 29;

else

return 28;

}

else if (month == 1 || month == 3 || month == 5 || month == 7 || month == 8

|| month == 10 || month == 12)

return 31;

else

return 30;

}

struct Date

{

int year;

int month;

int day;

};

int dayOfYear(Date date)

{

int daysNumb = 0;

for (int i = 1; i < date.month; i++)

{

daysNumb += monthLength(date.year, i);

}

return daysNumb + date.day;

}

int countLeapYears(Date d)

{

int years = d.year;

if (d.month <= 2)

years--;

return years / 4 - years / 100 + years / 400;

}

int daysBetween(Date d1, Date d2)

{

if (d1.year > d2.year || (d1.year == d2.year && d1.month > d2.month) || (d1.year == d2.year && d1.month == d2.month && d1.day > d2.day))

{

return -1;

}

else

{

long int n1 = d1.year \* 365 + d1.day;

for (int i = 0; i < d1.month - 1; i++)

n1 += monthLength(d1.year, i);

n1 += countLeapYears(d1);

long int n2 = d2.year \* 365 + d2.day;

for (int i = 0; i < d2.month - 1; i++)

n2 += monthLength(d2.year, i);

n2 += countLeapYears(d2);

return (n2 - n1);

}

}

int main()

{

Date since, till;

cout << "Enter first date (y m d): ";

cin >> since.year >> since.month >> since.day;

cout << "Enter second date (y m d): ";

cin >> till.year >> till.month >> till.day;

cout << daysBetween(since, till) << endl;

return 0;

}

3.6.3 (1) #include <iostream>

#include <ctime>

using namespace std;

int main()

{

time\_t t = time(NULL);

tm tl = \*localtime(&t);

cout << tl.tm\_year + 1900 << "-" << tl.tm\_mon + 1 << "-" << tl.tm\_mday << endl;

return 0;

}

3.8.7 (1) #include <iostream>

#include <ctime>

using namespace std;

bool isPrime(int num)

{

if (num <= 1)

return false;

for (int i = 2; i < num; i++)

if (num % i == 0)

return false;

return true;

}

int main()

{

for (int i = 0; i <= 21; i++)

if (isPrime(i))

cout << i << " ";

cout << endl;

return 0;

}

3.10.8 (1)

#include <iostream>

using namespace std;

void increment(int& val, int increm = 1)

{

val += increm;

}

int main(void)

{

int var = 0;

for (int i = 0; i < 10; i++)

if (i % 2)

increment(var);

else

increment(var, i);

cout << var << endl;

return 0;

}

3.10.8 (2) #include <iostream>

using namespace std;

void increment(int& val, int increm = 1)

{

val += increm;

}

void increment(float& val, float increm = 1)

{

val += increm;

}

int main(void)

{

int intvar = 0;

float floatvar = 1.5;

for (int i = 0; i < 10; i++)

if (i % 2) {

increment(intvar);

increment(floatvar, sqrt(intvar));

}

else {

increment(intvar, i);

increment(floatvar);

}

cout << intvar \* floatvar << endl;

return 0;

}