МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Тверской государственный университет»

Факультет прикладной математики и кибернетики

Направление 02.03.02 – Фундаментальная информатика и информационные технологии

Профиль подготовки «Информатика и компьютерные науки»

**Отчет по итогам учебной практики по получению первичных профессиональных умений и навыков**

2017 - 2018 уч. год, 1 семестр

**Автор**: студентка 1 курса

Николаева Мария Сергеевна

**Руководитель практики**:

старший преподаватель

Мальцева Людмила Федеровна

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тверь – 2017

**2. Задание 1  
Задача №16**Найти все такие простые числа, не превосходящие заданного N, двоичная запись которых представляет собой чередующуюся последовательность нулей и единиц.

#include <iostream>   
bool P(int n)   
{

bool f = true;   
 if (n % 2 == 0 && n!=2) return false;   
 for (int i = 3; i < n; i++)   
 {   
 if (n%i == 0)   
 {   
 f = false;   
 break;   
 }   
 }  
 return f;   
}

bool cher(int n)   
{   
 bool fl = true;   
 int i;   
 int nn = n;   
 int k = 0;   
 if (P(n))   
 {   
 while (nn > 0)   
 {   
 nn = nn / 2;   
 k++;   
 }   
 int \*p = new int[k];   
 for (i = 0; i < k; i++)   
 {   
 p[i] = n % 2;   
 n = n / 2;   
 } 4  
  
 for (i = 0; i < k - 1; i++)   
 {   
 if (p[i] != p[i + 1]) continue;   
 else   
 {   
 fl = false;   
 break;   
 }   
 }   
 delete p;   
 return fl;   
}   
else return false;   
}

int main()   
{   
 int n;   
 printf("Vvedite chislo\n");   
 scanf\_s("%d", &n);   
 printf("Otvet: ");   
 for (int i = 2; i < n; i++)   
 if (cher(i)) printf("%d, ", i);   
 system("pause");   
 return 0;   
}

**Функция Р: определяет, является ли число простым. Возвращает f**  
1) положить флаг f = true  
2) если число четное и не равно 2, положить f = false и перейти на шаг 6  
3) для i = 3 до n перейти на шаг 4  
4) если i делит n, положить f = false и перейти на шаг 5,   
 иначе перейти на шаг 3  
5) вернуть f   
6) конец выполнения функции

**Функция cher: определяет или нет, является ли двоичная запись числа** **чередующейся последовательностью нулей и единиц. Возвращает fl**  
1) положить fl = true, nn = n, k = 0  
2) если n – простое, перейти на шаг 3, иначе перейти на шаг 15  
 5  
3) пока nn > 0 перейти на шаг 4, иначе перейти на шаг 6  
4) положить nn = nn / 2  
5) положить k = k + 1  
6) объявить p – массив из k элементов  
7) для i = 0 до k перейти на шаг 8, иначе перейти на шаг 11  
8) положить p[i] = n%2  
9) положить n = n / 2  
10) перейти на шаг 7  
11) для i = 0 до k-1перейти на шаг 12, иначе перейти на шаг   
12) если p[i] ≠ p[i+1] перейти на шаг 11, иначе перейти на шаг 13   
13) положить fl = false и перейти на шаг 14  
14) освободить память, выделенную под p  
15) положить fl = false  
16) окончание работы функции  
  
**Функция main:**1) ввести n  
2) для i = 2 до n перейти на шаг 3, иначе перейти на шаг 4  
3) если cher(i) = true, вывести i, иначе перейти на шаг 2   
4) окончание работы программы

При входном n = 3 программа выведет 2, верно: 210 = 102.  
При входном n = 5 программа выведет 2.  
При входном n = 45 программа выведет 2, 5, верно: 210 = 102, 510 = 1012 .

**3. Задание 2**   
**Задача №12**  
Найти и распечатать строку заданной матрицы размером n\*n, в которой длина максимальной серии минимальна.  
  
[#include](https://vk.com/feed?section=search&q=%23include) <iostream>   
int\*\* create(int n)   
{   
 int \*\*a= new int \*[n];   
 for (int i = 0; i < n; i++)   
 a[i] = new int[n];   
 return a;   
}   
 6  
int\*\* fill(int\*\*a, int n)   
{   
 printf("Vvedite massiv\n");   
 for (int i = 0; i < n; i++)   
 for (int j = 0; j < n; j++)   
 scanf\_s("%d", &a[i][j]);    
 return a;   
}   
  
int ser(int\*\*a, int n, int i)   
{   
 int max=0, k=1;   
 for (int j = 0; j < n; j++)   
 {   
 if (a[i][j] == a[i][j + 1])   
 {   
 k++;   
 if (k>max) max = k;   
 }   
 else k = 1;   
 }   
 return max;   
 }   
void min(int\*\* a, int n, int &im)   
{   
 int min;   
 min = ser(a, n, 0);  
 for (int i = 1; i < n; i++)   
 if (min>ser(a, n, i))   
 {   
 min = ser(a, n, i);   
 im = i;   
 }   
 }   
  
int main()   
{   
 int n;   
 printf("Vvedite n\n");   
   
 7  
 scanf\_s("%d", &n);   
 int im = 0;   
 int \*\*a = create(n);   
 a = fill(a, n);   
 min(a, n, im);   
 printf("\n");   
 for (int j = 0; j < n; j++)   
 printf("%d ", a[im][j]);   
 for (int i=0; i<n; i++)   
 delete []a[i];    
 delete[]a;   
 printf("\n");   
 system("pause");   
 return 0;   
}

**Функции create и fill – выделение памяти под матрицу и ее заполнение**.

**Функция ser: поиск максимальной серии в строке**. **Возвращает max.**  
1) положить max = 0, k = 1  
2) для j = 0 до n перейти на шаг 3, иначе перейти на шаг 7  
3) если a[i][j] = a[i][j+1], перейти на шаг 4, иначе перейти на шаг 6  
4) k = k + 1  
5) если k > max, положить max = k и перейти на шаг 2  
6) положить k = 1 и перейти на шаг 2   
7) конец работы функции

**Функция min: определяет индекс строки im с наименьшей длиной максимальной серии.**1) положить min = ser(a, n, 0)  
2) для i = 1 до n перейти на шаг 3, иначе перейти на шаг 6  
3) если ser(a, n, i) < min, перейти на шаг 4, иначе перейти на шаг 2  
4) положить min = ser(a, n, i)  
5) положить im = i, перейти на шаг 2

**Функция main:**1) ввести n  
2) выделить память под матрицу и заполнить ее  
3) найти im с помощью вызова функции min(a, n, im)  
4) вывести строку номер im матрицы  
  
 8  
5) освободить память, выделенную под матрицу  
6) конец работы программы

При входных n = 4   
 и матрице   
 1 1 1 3  
 2 3 3 4  
 5 6 7 8  
 8 8 8 8   
программа выведет 5 6 7 8, верно: длина максимальной серии в этой строке равна 1, и это минимальная из максимальных серий всех строк.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1000 *стек* | ***n*** 4 | ***im*** 0 | ***\*\*a*** 3000 | ***\*\*a***3000 | ***i*** 0…n-1 | ***n*** 4 |
| ***\*\*a*** 3000 | ***n*** 4 | ***i*** 0…n-1 | ***j*** 0…n-1 | ***\*\*a***3000 | ***n*** 4 |
| ***min*** 3 | ***\*\*a***3000 | ***n*** 4 | ***i***  0 | ***max*** 0123 | ***k*** 1 2 3 |
| ***j*** 0…n-1 | ***i***  0…n-1 | ***\*\*a***3000 | ***n*** 4 | ***i***  i | ***j*** 0…n-1 |
|  |  |  |  |  |  |
| 3000 *дина- мичес- кая память* | 3000   1 | 3004  2 | 3008  5 | 300C  8 | 3010  1 | 3014  1 |
| 3018  3 | 301C  3 | 3020  3 | 3024  4 | 3028 6 | 302C  7 |
| 3030  8 | 3034  8 | 3038  8 | 303C  8 |  |  |

При входных n = 5   
 и матрице   
 1 1 1 3 6  
 2 3 3 4 4  
 5 6 7 8 8   
 8 8 8 8 9   
 1 1 3 4 5  
программа выведет 2 3 3 4 4, верно: в этой строке максимальная длина серии равна 2. Такая же длина также в третьей и пятой строках матрицы; программа выводит первую такую строку.

9