//задача 1

Напишите на языке Си функцию, которая переставляет в обратном порядке элементы массива. Параметры функции: 1) указатель на массив целых чисел, 2) количество элементов в массиве. Тип возвращаемого подпрограммой значение: void.

void fun(int \*a, int n)

{

for (int i = 0; i <n/2; i++)

{

int x = a[i];

a[i] = a[n - 1 - i];

a[n - 1 - i] = x;

}

}

//задача 2

Напишите на языке Си функцию, которая находит разницу между максимальным и минимальным элементами в массиве. Параметры функции: 1) указатель на массив целых чисел, 2) количество элементов в массиве. Возвращаемое подпрограммой значение: целое число, которое равно разнице между максимальным и минимальным элементами в массиве.

float f\_raznica(int \* m, int pmas)

{

float maxel=0;

float minel;

float rez;

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

{

if (m[i] > maxel)

{

maxel = m[i];

}

minel = maxel;

}

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

{

if (m[i] < minel)

{

minel = m[i];

}

}

rez = maxel - minel;

return (rez);

}

// задача 3

Напишите на языке Си функцию, которая находит разницу между четными и нечетными элементами в массиве. Параметры функции: 1) указатель на массив целых чисел, 2) количество элементов в массиве. Возвращаемое подпрограммой значение: целое число, которое равно разнице между четными и нечетными элементами в массиве.

int difference(int\* A, int Length)

{

int Seven=0, Sodd=0;

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

if (i % 2 == 0)

Seven = Seven + A[i];

else

Sodd = Sodd + A[i];

return Seven - Sodd;

}// задача 4

Напишите на языке Си функцию, которая находит разницу между элементами стоящими на четных и нечетных позициях в массиве. Параметры функции: 1) указатель на массив целых чисел, 2) количество элементов в массиве. Возвращаемое подпрограммой значение: целое число, которое равно разнице между элементами стоящими на четных и нечетных позициях в массиве.

int fun2(int \*a, int n)

{

int sum1, sum2;

sum1 = sum2 = 0;

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

{

if (i%2==1)

sum1 += a[i];

else

sum2 += a[i];

}

return sum2 - sum1;

}

// 5 задача

Напишите на языке Си функцию, которая находит индекс элемента в массиве, значение которого наиболее близко к заданному целому числу. Параметры функции: 1) указатель на массив вещественных чисел, 2) количество элементов в массиве, 3) искомое целое число. Возвращаемое подпрограммой значение: целое число, которое равно индексу наиболее близкого элемента массива.

int fun3(float \*a, int n, int b)

{

int k=0;

float buff = abs(a[0] - b);

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

{

if (abs(a[i] - b) < buff)

{

buff = abs(a[i] - b);

k = i;

}

}

return k;

}

// 6 задача

Напишите на языке Си функцию, которая получает целочисленный массив и целое число, а определяет индекс элемента массива, который больше всего отличается по величине от заданного числа и сам является при этом четным. Параметры функции: 1) указатель на массив целых чисел 2) число элементов в массиве 3) целое число. Возвращаемое функцией значение: индекс найденного элемента или -1, если требуемого элемента нет.

int fun4(int \*a, int n, int b)

{

int k = 0;

int buff = abs(a[0] - b);

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

{

if( (a[i] % 2)!= 1)

{

if (abs(a[i] - b) > buff)

{

buff = abs(a[i] - b);

k = i;

}

}

}

return k;

}

// 7 задача

Напишите на языке Си функцию, которая находит сумма каких элементов (первая половина или втораяполовина) больше. Параметры функции: 1) указатель на массив целых чисел, 2) количество элементов в массиве. Возвращаемое подпрограммой значение: целое число, которое равно 0 если если сумма первой половины больше, 1 если сумма второй половины больше. Если количество элементов – нечетное, центральный элемент не суммируется.

int fun5(int \*a, int n)

{

int sum1, sum2;

sum1 = sum2 = 0;

if (n % 2 == 1)

{

n--;

a[(n / 2)] = 0;

}

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

{

if (i < n / 2)

sum1 += a[i];

if (i > n / 2)

sum2 += a[i];

}

sum2 += a[n];

if (sum1 > sum2)

return 1;

return 0;

}

// 8 задача

Напишите на языке Си функцию, которая ищет в массиве самую длинную непрерывную последовательность чисел, имеющих заданное значение. Параметры функции: 1) указатель на массив целых чисел, 2) количество элементов в массиве, 3) искомое значение. Возвращаемое подпрограммой значение: беззнаковое целое число, которое равно длине самой длинной непрерывной последовательности чисел, имеющих заданное значение.

int fun6(int \*a, int n, int k)

{

int col = 0, maxcol=0;

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

{

if (a[i] == k) col++;

if (a[i] != k)

{

if(col>maxcol)

maxcol = col;

col = 0;

}

}

return maxcol;

}

// 9 задача

Напишите на языке Си функцию, которая находит самую длинную неубывающую последовательность чисел в массиве. Параметры функции: 1) указатель на массив целых чисел, 2) количество элементов в массиве. Возвращаемое подпрограммой значение: беззнаковое целое число, которое равно длине самой длинной неубывающей последовательности элементов массива.

int fun7(int \*a, int n)

{

int col = 1, maxcol = 0;

for (int i = 1; i < n; i++)

{

if (a[i]>=a[i-1]) col++;

if (a[i]<a[i-1])

{

if (col>maxcol)

maxcol = col;

col = 0;

}

}

return maxcol;

}

// 10 задача

Напишите на языке Си функцию, которая получает массив чисел, каждое из которых не превосходит 100, и подсчитывает в нем количество элементов, являющихся степенями числа 2 (2, 4, 8, 16, . . . ). Параметры функции: 1) указатель на массив целых чисел 2) число элементов в массиве. Возвращаемое функцией значение: число найденных элементов.

int fun8(int \*a, int n)

{

int col = 0;

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

{

switch (a[i])

{

case 2: col++;

break;

case 4: col++;

break;

case 8: col++;

break;

case 16: col++;

break;

case 32: col++;

break;

case 64: col++;

break;

}

}

return col;

}

//11 задача

Напишите на языке Си функцию, которая сортирует любым способом элементы массива по невозрастанию. Параметры функции: 1) указатель на массив беззнаковых целых чисел, 2) количество элементов в массиве. Возвращаемое подпрограммой значение: void. Библиотечные функции сортировки использовать запрещается.

void fun9(int \*a, int n)

{

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

{

for (int j = 0; j < n - 1 - i; j++)

{

if (a[j] < a[j + 1])

{

int x = a[j + 1];

a[j + 1] = a[j];

a[j] = x;

}

}

}

}

//12 задача

Напишите на языке Си функцию, которая получает квадратную матрицу вещественных чисел (массив) и транспонирует матрицу. Параметры функции: 1) указатель на массив вещественных чисел 2) число элементов в строке(столбце). Возвращаемое функцией значение: void.

void fun10(int a[3][3], int n)

{

int t;

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

{

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

{

t = a[i][j];

a[i][j] = a[j][i];

a[j][i] = t;

}

}

}

//13 задача

Напишите на языке Си функцию, которая получает квадратную матрицу вещественных чисел (массив) и находит сумму элементов на главной диагонали. Параметры функции: 1) указатель на массив вещественных чисел 2) число элементов в строке(столбце). Возвращаемое функцией значение: сумма элементов главной диагонали.

int fun11(int a[3][3], int n)

{

int sum = 0;

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

{

sum += a[i][i];

}

return sum;

}

//14 задача

Напишите на языке Си функцию, которая получает квадратную матрицу вещественных чисел (массив) и заменяет нулевые элементы на максимальный элемент матрицы. Параметры функции: 1) указатель на массив вещественных чисел 2) число элементов в строке(столбце). Возвращаемое функцией значение: количество замен.

int SquareMatrix(int A[3][3], int Length)

{

int MaxElement = A[0][0], Count=0;

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

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

if (MaxElement < A[i][j])

MaxElement = A[i][j];

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

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

if (A[i][j] == 0)

{

A[i][j] = MaxElement;

Count++;

}

return Count;

}

//15 задача

Напишите на языке Си функцию, которая получает 2 квадратную матрицу вещественных чисел (массивы) и находит их сумму. Параметры функции: 1) указатель на первый массив вещественных чисел2) указатель на второй массив вещественных чисел3) число элементов в строке(столбце). Тип возвращаемого функцией значения: void. Результат суммирования помещать в первой матрице.

void fun13(int a[3][3], int b[3][3], int n)

{

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

{

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

{

a[i][j] += b[i][j];

}

}

}

//16 задача

Напишите на языке Си функцию, которая получает строку символов (с нулевым ограничителем ) и располагает все символы в обратном порядке. Параметр функции: указатель на строку. Возвращаемое функцией значение: void.

void fun14(char \*c)

{

for (int i = 0; i < strlen(c)/2; i++)

{

int x = c[i];

c[i] = c[strlen(c) - 1 - i];

c[strlen(c) - 1 - i] = x;

}

}

// 17 задача

Напишите на языке Си функцию, которая получает строку символов (с нулевым ограничителем ) и циклически сдвигает все элементы на одну позицию влево (первый элемент становится последним, второй первым и т.п.). Параметр функции: указатель на строку. Возвращаемое функцией значение: void.

void fun15(char \*c)

{

char a = c[0];

for (int i = 1; i < strlen(c); i++)

{

char t = c[i];

c[i] = c[i - 1];

c[i - 1] = t;

}

c[strlen(c)-1] = a;

}

//18 задача

Напишите на языке Си функцию, которая получает строку символов (с нулевым ограничителем ) и меняет все буквы ‘x’ на ‘y’, ‘X’ на ‘Y’. Параметр функции: указатель на строку. Возвращаемое функцией значение: число сделанных замен.

int fun16(char \*c)

{

int col1 = 0;

for (int i = 0; i < strlen(c); i++)

{

if (c[i] == 'x')

{

c[i] = 'y';

col1++;

}

if (c[i] == 'X')

{

c[i] = 'Y';

col1++;

}

}

return col1;

}

//19 задача

Напишите на языке Си функцию, которая получает строку символов (с нулевым ограничителем ) и заменяет в ней все символы цифр символом #, а символы кириллицы – символом $. Параметр функции: указатель на строку. Возвращаемое функцией значение: число сделанных замен.

int CharChanger(char\* Str)

{

int Count=0;

for (int i = 0; Str[i] != '\0'; i++)

{

if (Str[i] >= '0' && Str[i] <= '9')

{

Str[i] = '#';

Count++;

}

if ((Str[i] >= 'а' && Str[i] <= 'я') || (Str[i] >= 'А' && Str[i] <= 'Я'))

{

Str[i] = '$';

Count++;

}

}

return Count;

//20 задача

Напишите на языке Си функцию, которая получает строку символов (с нулевым ограничителем ) и удаляет все повторяющиеся символы, идущие подряд. Параметр функции: указатель на строку. Возвращаемое функцией значение: число сделанных удалений.

int fun18(char \*c)

{

int n = strlen(c), col=0, d=1;

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

{

d = 0;

if (c[i] == c[i + 1])

{

d = 1;

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

{

c[j] = c[j + 1];

}

n--;

col++;

}

}

return col;

}

//задача 21

Напишите на языке Си функцию, которая получает строку символов (с нулевым ограничителем ) и убирает в ней все начальные и все конечные пробелы, если таковые имеются. Параметр функции: указатель на строку. Возвращаемое функцией значение: строка без пробелов.

void del(char \* str)

{

for (int i=0; i<strlen(str); i++ )

{

i = 0;

if (str[i] == ' ')

{

for (int j=0; j < strlen(str); j++)

{

str[j] = str[j + 1];

}

}

else

break;

}

for (int i = strlen(str) - 1; i>=0; --i)

{

if (str[i] == ' ')

{

str[i] = '\0';

}

else

break;

}

}

//задача 22

Напишите на языке Си функцию, которая получает строку символов (с нулевым ограничителем ) и проверяет, является ли она симметричной (читается одинаково справа налево и слева направо). Параметр функции: указатель на строку. Возвращаемое функцией значение: 1 если строка симметрична, 0 – в противном случае

int fun20(char \*c)

{

for (int i = 0; i < strlen(c); i++)

{

if (c[i] != c[strlen(c) - 1 - i])

{

return 0;

break;

}

}

return 1;

}

//задача 23

Номер мобильного телефона является 10-символьной комбинацией цифр, которая может начинаться с цифры 8, либо с +7, либо не содержать никакого начального символа вообще. Напишите на языке Си функцию, которая получает строку (с нулевым ограничителем) с номером мобильного телефона любого из указанных форматов и выделяет из него номер сети – первые три цифры 10-символьной комбинации. Параметры функции: указатель на строку. Возвращаемое функцией значение: строка номера сети (с нулевым ограничителем).

char fun21(char \*c)

{

char b[3];

if (char(c[0] == '8'))

{

for (int i = 0; i < 3; i++)

b[i] = c[i + 1];

return \*b;

}

if (char(c[0] == '+'))

{

b[0] = c[2];

b[1] = c[3];

b[3] = c[4];

return \*b;

}

b[0] = c[0];

b[1] = c[1];

b[2] = c[2];

return \*b;

}

//задача 24

Напишите на языке Си функцию, которая получает две даты в формате дд-мм-гггг и определяет, сколько дней отделяет одну дату от другой. Параметры функции: 1) указатель на строку первой даты, 2) указатель на строку второй даты. Возвращаемое функцией значение: целое число, которое равно числу дней.

int fun22(char \*c, char \*x)

{

int a = 0, b = 0, c1 = 0, a1 = 0, b1 = 0, c2 = 0;

sscanf(c, "%d-%d-%d", &a, &b, &c1);

sscanf(x, "%d-%d-%d", &a1, &b1, &c2);

int col = abs(a - a1) + abs(b - b1) \* 30 + abs(c1 - c2) \* 365;

return col;

}

// задача 25

Напишите на языке Си функцию, которая получает дату как строку символов (с нулевым ограничителем ) в формате ДД-ММ-ГГГГ и преобразует ее в строку «ДД Месяца ГГГГ года». Например, «01-15-2017» должно быть преобразовано в «01 мая 2017 года» Параметр функции: указатель на исходную строку. Возвращаемое функцией значение: преобразованная строка.

char fun23(char \*c)

{

char b[30];

char mes[30];

int day = 0, mon = 0, yaer = 0;

sscanf(c, "%d-%d-%d", &day, &mon, &yaer);

switch (mon)

{

case 1: char mes[] = "января";

break;

case 2: char mes[]= " февраля";

break;

case 3: char mes[] = " марта";

break;

case 4: char mes[] = " апреля";

break;

case 5: char mes[] = " мая";

break;

case 6: char mes[] = " июня";

break;

case 7: char mes[] = " июля";

break;

case 8: char mes[] = " августа";

break;

case 9: char mes[] = " сентября";

break;

case 10: char mes[] = " октября";

break;

case 11: char mes[] = " ноября";

break;

case 12: char mes[] = " декабря";

break;

}

sprintf(b, "%d", day);

for(int i=0; i<strlen(mes); i++)

sprintf(b, "%d %d года ", day, yaer);

}