**Task 1:**

Write a program in C++ that reads integers from a file into an array (dynamically created) and shifts all even numbers in ascending order to the left and odd numbers in descending order to the right side.

**Sample Output:**

Updated Data:2 8 10 14 82 21 15 9 3 1

14 2 3 15 1 82 8 9 10 21 55 12 14 16 18

**data.txt**

**Note:** Use only one array and make necessary and required changes to it only.

#include<iostream>

#include<fstream>

using namespace std;

void write(ofstream&fout)

{

fout.open("data.txt");

if (fout.is\_open())

{

fout << "14 2 3 15 1 82 8 9 10 21 55 12 14 16 18";

}

fout.close();

}

int getlength(ifstream&fin)

{

int a;

int x = 0;

fin.open("data.txt", ios::in);

if (!fin)

{

cout << "NO FILE FOUND" << endl;

}

else

{

while (!fin.eof())

{

fin >> a;

x++;

}

}

fin.close();

return x;

}

int\*mdynamic(int s)

{

int\*array = new int[s];

return array;

}

void read(ifstream&fin, int\*ptr, int size)

{

int value;

int index1 = 0;

int index2 = 0;

fin.open("data.txt", ios::in);

if (!fin)

{

cout << "File not found" << endl;

}

else

{

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

{

fin >> value;

if (value % 2 == 0)

{

ptr[index1] = value;

index1++;

}

}

index2 = index1;

for (int i = 0; i < index1 - 1; i++)

{

for (int k = i; k < index1 - 1; k++)

{

if (ptr[i]>ptr[k + 1])

{

int temp = ptr[i];

ptr[i] = ptr[k + 1];

ptr[k + 1] = temp;

}

}

}

}

fin.close();

fin.open("data.txt", ios::in);

if (!fin)

{

cout << "File not found" << endl;

}

else

{

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

{

fin >> value;

if (value % 2 != 0)

{

ptr[index1] = value;

index1++;

}

}

for (int i = index2; i < index1 - 1; i++)

{

for (int k = i; k < index1 - 1; k++)

{

if (ptr[i]<ptr[k + 1])

{

int temp = ptr[i];

ptr[i] = ptr[k + 1];

ptr[k + 1] = temp;

}

}

}

cout << "Updated Data: ";

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

{

cout << ptr[i] << " ";

}

cout << endl;

}

fin.close();

}

int main()

{

ofstream fout;

ifstream fin;

write(fout);

int size = getlength(fin);

int\*array = NULL;

array = mdynamic(size);

read(fin, array, size);

return 0;

}

**Task 2:**

Write a program in C++ that reads data from user into a dynamically growing array until user enters 0. Separate 1-digit and 2-digit numbers from the given data and store them in separately growing dynamic arrays.

**Sample Output:**

Input Data: 2 3 15 1 82 8 9 10 21 55 12 14 16 18 0

One-digit: 2 3 1 8 9

Two-digit: 15 82 10 21 55 12 14 16 18

#include<iostream>

using namespace std;

void copy(int\*old, int\*n, int s)

{

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

{

n[i] = old[i];

}

}

int\*regrow(int\*old, int&s, int v)

{

int\*newarray = new int[s + 1];

copy(old, newarray, s);

newarray[s] = v;

s++;

delete[]old;

old = NULL;

return newarray;

}

int \*regrow1(int\*old, int&s1, int v)

{

int\*one = new int[s1 + 1];

copy(old, one, s1);

one[s1] = v;

s1++;

delete[]old;

old = NULL;

return one;

}

int main()

{

int size = 0, size1 = 0, value = 0;

cout << "Enter values: ";

cin >> value;

int\*data = new int[1];

data[0] = value;

size++;

while (cin >> value && value != 0)

{

data = regrow(data, size, value);

}

int\*one = new int[1];

int\*two = new int[1];

int value1;

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

{

if (data[j] >= 0 && data[j] <= 9)

{

value1 = data[j];

one = regrow1(one, size1, value1);

}

}

cout << "One-digit: ";

for (int i = 0; i < size1; i++)

{

cout << one[i] << " ";

}

cout << endl;

size1 = 0;

for (int k = 0; k < size; k++)

{

if (data[k] > 9)

{

value1 = data[k];

two = regrow1(two, size1, value1);

}

}

cout << "Two-digit: ";

for (int i = 0; i < size1; i++)

{

cout << two[i] << " ";

}

cout << endl;

return 0;

}