```
qin@linprog8.cs.fsu.edu:~>clear
```

```
qin@linprog8.cs.fsu.edu:~>~vastola/usub/submit1 arrays.cpp
********************* File submitted *********************
   Here are the contents of your submitted file:
**********************************************************
/* Name: Caijun Qin
Date: 10/29/2018
Section: 6
Assignment: 5
Due Date: 11/06/2018
About this project: This program provides a selection of editing functions that
programmers can use to modify arrays when writing code including arrays. Namely,
the user can fill the array with random values, insert or delete a new element,
reverse the order of elements, find the maximum odd value, or simply print out
the array.

Assumptions: The user only inputs integers and valid capital letters. The user
will choose menu option F to fill the array first before any other operation.
All work below was performed by Caijun Qin */

#include <iostream>
#include <ctime>
#include <cstdlib>
using namespace std;

/*FUNCTION DECLARATIONS*/
void FillArray(int array[], const int SIZE, int min, int max);
void Insert(int array[], const int SIZE, int value, int index);
void Delete(int array[], const int SIZE, int index);
void Reverse(int array[], const int SIZE);
int MaxOdd(const int array[], const int SIZE);
void PrintArray (const int arr[], const int size);
void menu();

int main(){
    //variables
    const int SIZE = 15;
    int list[SIZE] = {};
    char menuSelect = '0';

    //starts the program seen on screen
    menu();
    do{
        cout << "\n Enter your menu selection:  ";
        cin >> menuSelect;

        if(menuSelect == 'P'){
            PrintArray(list, SIZE);
        } else if(menuSelect == 'F'){
            //variables
            int min = 0;
            int max = 0;

            cout << "What is the minimum random value?  ";
            cin >> min;
            cout << "What is the maximum random value?  ";
            cin >> max;
            FillArray(list, SIZE, min, max);

        } else if(menuSelect == 'I'){
            //variables
            int insert = 0;
            int index = 0;

            cout << "Enter value to insert: ";
            cin >> insert;
            cout << "Enter index at which to insert:  ";
            cin >> index;
            Insert(list, SIZE, insert, index);
            PrintArray(list, SIZE);

        } else if(menuSelect == 'D'){
            //variables
            int del = 0;

            cout << "Enter index of item to delete:  ";
            cin >> del;
            Delete(list, SIZE, del);
            PrintArray(list, SIZE);

        } else if(menuSelect == 'R'){
            Reverse(list, SIZE);
            PrintArray(list, SIZE);

        } else if(menuSelect == 'X'){
            cout << "The maximum odd value in the array is " << MaxOdd(list, SIZ
E);
            cout << "\n";

        } else if(menuSelect == 'M'){
            menu();

        } else if(menuSelect != 'Q'){
            cout << "Invalid selection. Please try again. \n";
        }
    } while (menuSelect != 'Q');

    return 0;
}

/*FUNCTION DEFINITIONS*/
//FUNCTION 1
void FillArray(int array[], const int SIZE, int min, int max){
        //plants the seed value
        srand(time(0));

        //generates the sign of the number, if min falls below 0
        int sign = -1;

        //classifies the random number into nonnegative or both signs allowed
        int pNum = 0;
        int pnNum = 0;

        //splits up random number generation task based on range of choice
        //min is mistakenly greater than max
        if(max < min){
            char switchMaxMin = 'N';
            cout << "The upper bound of the range must be greater than the ";
            cout << "lower bound. \n";
            cout << "Do you want to switch the max and min?  Enter Y or N:  ";
            cin >> switchMaxMin;
            cout << "\n";

            //switches the max and min
            if(switchMaxMin == 'Y' || switchMaxMin == 'y'){
                int temporary = max;
                max = min;
                min = temporary;
            }
        }

        //min is the same value as max, with the range being 0 in size
        if(min == max){
```

```cpp
            for(int counter = 0; counter < SIZE; counter++){
                array[counter] = min;
            }
        }

        //min is nonnegative and max is positive
        if(min >=0 && max > 0 && min != max){
            for(int counter = 0; counter < SIZE; counter++){
                pNum = min + (rand() % ((max - min) + 1));
                array[counter] = pNum;
            }

        }

        //min is negative while max is positive
        if(min < 0 && max > 0) {
            for(int counter = 0; counter < SIZE; counter++){
                sign = rand() % 2;
                if(sign == 0){
                    pnNum = -(rand() % (abs(min) + 1));
                } else if(sign == 1){
                    pnNum = rand() % (max + 1);
                }
                array[counter] = pnNum;
            }
        }

        //min is negative and max is at most 0
        if(min < 0 && max <= 0  && min != max){
            for(int counter = 0; counter < SIZE; counter++){
                pnNum = -(abs(max) + (rand() % (abs(max - min) + 1)));
                array[counter] = pnNum;
            }
        }

        return;
}

//FUNCTION 2
void Insert(int array[], const int SIZE, int value, int index){
    //variables
    int lastIndex = SIZE - 1;

    //computation of pushing each entry up one since the index of insertion
    for(int counter = lastIndex; counter > index; counter--){
        array[counter] = array[counter - 1];
    }
    array[index] = value;

    return;
}
```

```
//FUNCTION 3
void Delete(int array[], const int SIZE, int index){
    //variables
    int lastIndex = SIZE - 1;

    //moves each entry at or to the right of index one position leftwards
    for(int counter = index; counter < lastIndex; counter++){
        array[counter] = array[counter + 1];
    }
    array[lastIndex] = 0;

    return;
}

//FUNCTION 4
void Reverse(int array[], const int SIZE){
    //variables
    int lastIndex = SIZE - 1;
    //temporary placeholder
    int temp = 0;

    //inverts position of each entry
    if(SIZE % 2 == 0){
        for(int counter = 0; counter < SIZE; counter++){
            //the mirror position of an entry
            int mirrorPos = lastIndex - counter;

            //swaps the two entries
            temp = array[mirrorPos];
            array[mirrorPos] = array[counter];
            array[counter] = temp;
        }
    } else if(SIZE % 2 == 1){
        for(int counter = 0; counter < lastIndex / 2; counter++){
            //the mirror position of an entry
            int mirrorPos = lastIndex - counter;

            //swaps the two entries
            temp = array[mirrorPos];
            array[mirrorPos] = array[counter];
            array[counter] = temp;
        }
    }

    return;
}

//FUNCTION 5
int MaxOdd(const int array[], const int SIZE){
    //variables
    int currentOdd;
    //checks each odd number consecutively to compare size
    for(int counter = 0; counter < SIZE; counter++){
        if(abs(array[counter]) % 2 == 1 && array[counter] > currentOdd){
            currentOdd = array[counter];
        }
    }
    return currentOdd;
}

/* Definition of PrintArray*/
//PrintArray Function
//This function prints the contents of any interger array of any size seperated
by commas
void PrintArray(const int arr[], const int size)
{
   cout << "\nThe array:\n { ";
   for (int i = 0; i < size-1; i++)     // for loop, prints each item (not last
due to comma handling)
      cout << arr[i] << ", ";

   cout << arr[size-1] << " }\n";       // prints last item , sans comma
}

/*The Menu Loop*/
//this merely prints out the menu
void menu(){
    cout << "\t ** Given features ** \n";
    cout << "P \t Print the array contents \n\n";
    cout << "\t ** Function Tests ** \n";
    cout << "F \t Fill the array with random values \n";
    cout << "I \t Insert \n";
    cout << "D \t Delete \n";
    cout << "R \t Reverse \n";
    cout << "X \t Max Odd Value \n\n";

    cout << "M \t Print this menu \n";
    cout << "Q \t Quit this program \n\n";

    return;
}




********************* END FEEDBACK ***********************
*    PLEASE VIEW ABOVE TO VERIFY THE CONTENTS OF         *
*              YOUR SUBMITTED FILE                       *
**********************************************************
qin@linprog8.cs.fsu.edu:~>
```