Muhammad Ahmer Zaman

18F-0336

ASSIGNMENT 2

**QUESTION 1**

**CODE:**

#include<iostream>

using namespace std;

void reverseString(char str[], int i)

{

cout << str[i];

if (i == 0)

{

return;

}

i--;

reverseString(str, i);

}

int main()

{

char str[100];

cout << "Enter a string:";

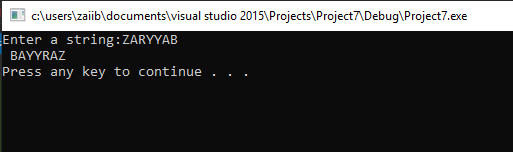
cin >> str;

int i = strlen(str);

reverseString(str, i);

cout << endl;

system("pause");

}

**QUESTION 2**

**CODE:**

#include<iostream>

using namespace std;

int factor(int n1, int n2)

{

if (n2 != 0)

return factor(n2, n1 % n2);

else

return n1;

}

int main()

{

int n1, n2;

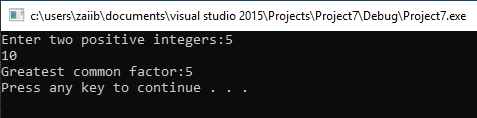
cout << "Enter two positive integers:";

cin >> n1 >> n2;

cout << "Greatest common factor:"<< factor(n1, n2) << endl;

system("pause");

}



**QUESTION 3**

**CODE:**

#include<iostream>

using namespace std;

int\* sort(int\* arr, int size)

{

int temp;

if (size == 1)

{

return arr;

}

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

{

if (arr[i] > arr[i + 1])

{

temp = arr[i];

arr[i] = arr[i + 1];

arr[i + 1] = temp;

}

}

sort(arr, size - 1);

}

int main()

{

int size;

cout << "Enter size of array:";

cin >> size;

int \*arr = new int[size];

cout << "Enter elements of array:";

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

{

cin >> \*(arr + i);

}

sort(arr, size);

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

{

cout << \*(arr + i);

if (i != size - 1)

cout << " , ";

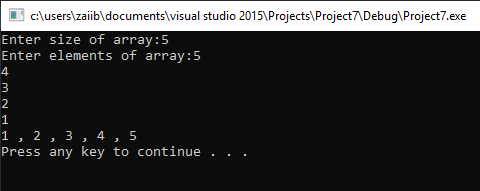
}

delete[]arr;

arr = NULL;

cout << endl;

system("pause");

}

**QUESTION 4**

**CODE:**

#include<iostream>

using namespace std;

bool PalRecursive(char str[], int s, int e)

{

if (s == e)

{

return true;

}

if (str[s] != str[e])

{

return false;

}

if (s < e + 1)

{

return PalRecursive(str, s + 1, e - 1);

}

return true;

}

bool Palindrome(char str[])

{

int n = strlen(str);

if (n == 0)

{

return true;

}

return PalRecursive(str, 0, n - 1);

}

int main()

{

char str[100];

cout << "Enter a string:";

cin >> str;

if (Palindrome(str))

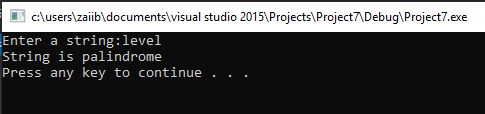
cout << "String is palindrome";

else

cout << "String is not palindrome";

cout << endl;

system("pause");

}

**QUESTION 5**

**CODE:**

#include<iostream>

using namespace std;

class employee {

public:

char name[25];

int idNum;

char department[25];

char position[25];

employee()

{

strcpy\_s(name, "");

idNum = 0;

strcpy\_s(department, "");

strcpy\_s(position, "");

}

employee(char\*nameMain, int idNumMain, char\* departmentMain, char\*positionMain)

{

strcpy\_s(name, nameMain);

idNum = idNumMain;

strcpy\_s(department, departmentMain);

strcpy\_s(position, positionMain);

}

void setData(char\* employeeName, int employeeIdNum, char\* employeeDepartment, char\* employeePosition)

{

strcpy\_s(name, employeeName);

idNum = employeeIdNum;

strcpy\_s(department, employeeDepartment);

strcpy\_s(position, employeePosition);

}

char\* getName()

{

return name;

}

int getIdNum()

{

return idNum;

}

char\* getDepartment()

{

return department;

}

char\* getPosition()

{

return position;

}

~employee()

{

cout << "I'm destructor!";

}

};

int main()

{

employee entity;

cout << "Enter name of the employee:";

cin.getline(entity.name, 25);

cout << "Enter ID number:";

cin >> entity.idNum;

cout << "Enter department:";

cin >> entity.department;

cout << "Enter position:";

cin >> entity.position;

entity.setData(entity.name, entity.idNum, entity.department, entity.position);

cout << "Name: " << entity.getName() << endl

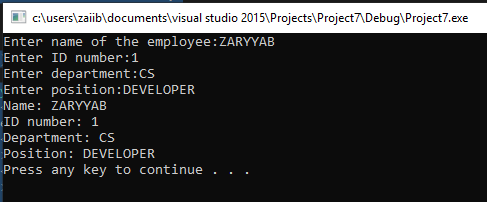
<< "ID number: " << entity.getIdNum() << endl

<< "Department: " << entity.getDepartment() << endl

<< "Position: " << entity.getPosition() << endl;

system("pause");

}



**QUESTION 6**

**CODE:**

#include<iostream>

using namespace std;

class Person {

public:

char name[25];

int age;

char gender;

char occupation[25];

char cooking;

Person()

{

strcpy\_s(name, "");

age = 0;

gender = 'm';

strcpy\_s(occupation, "");

cooking = 'n';

}

char\* getName()

{

return name;

}

void setName(char\* nameM)

{

strcpy\_s(name, nameM);

}

int getAge()

{

return age;

}

void setAge(int ageM)

{

age = ageM;

}

bool isMale()

{

if (gender == 'm')

return true;

else

return false;

}

void isFemale(char genderM)

{

gender = genderM;

}

char\* getOccupation()

{

return occupation;

}

void setOccupation(char\* occupationM)

{

strcpy\_s(occupation, occupationM);

}

bool canCook()

{

if (cooking == 'y' || cooking == 'Y')

return true;

else

return false;

}

void setCook(char cookingM)

{

cooking = cookingM;

}

~Person()

{

cout << "I'm destructor!";

}

};

int main()

{

Person p1;

cout << "Enter name of the person:";

cin.getline(p1.name, 25);

p1.setName(p1.name);

cout << "Enter age:";

cin >> p1.age;

p1.setAge(p1.age);

cout << "Is person Male or Female (M/F):";

cin >> p1.gender;

p1.isFemale(p1.gender);

cout << "Enter occupation:";

cin.ignore();

cin.getline(p1.occupation, 25);

p1.setOccupation(p1.occupation);

cout << "Can person cook (Y/N): ";

cin >> p1.cooking;

p1.setCook(p1.cooking);

cout << endl << endl << "Name:" << p1.getName() << endl

<< "Age:" << p1.getAge() << endl;

if (p1.isMale())

cout << "Person is Male" << endl;

else

cout << "Person is Female" << endl;

cout << "Occupation: " << p1.getOccupation() << endl;

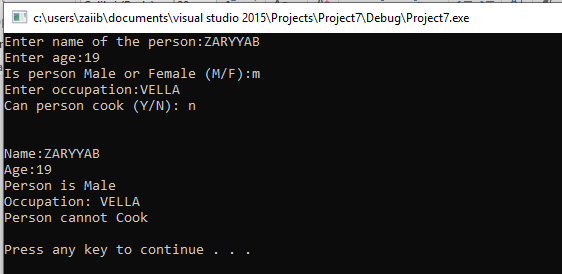
if (p1.canCook())

cout << "Person can Cook" << endl << endl;

else

cout << "Person cannot Cook" << endl << endl;

system("pause");

}

**QUESTION 7**

**CODE:**

#include<iostream>

using namespace std;

int Consonant(char\* arr, int size, int count)

{

if (size == 0)

{

return count;

}

if (arr[size] != '\0'&&arr[size] != 'a'&&arr[size] != 'A'&& arr[size] != 'e'&& arr[size] != 'E'&& arr[size] != 'i'&& arr[size] != 'I'&& arr[size] != 'o'&& arr[size] != 'O'&& arr[size] != 'u'&& arr[size] != 'U')

count++;

Consonant(arr, size - 1, count);

}

int main()

{

char arr[50];

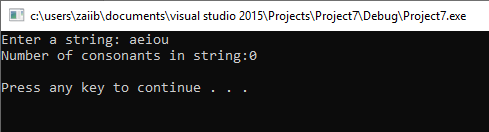
int count = 0;

cout << "Enter a string: ";

cin >> arr;

cout << "Number of consonants in string:" << Consonant(arr, strlen(arr), count) << endl << endl;

system("pause");

}