**TASK 1:**

#include<iostream>

using namespace std;

struct node

{

int data;

node\* prev, \* next;

};

class double\_link

{

int size;

node\* head;

node\* tail;

public:

double\_link()

{

size = 0;

head = NULL;

}

void insert(int data)//function insert the value in linked list

{

node\* temp = new node;

temp->data = data;

temp->prev = NULL;

temp->next = NULL;

if (head == NULL)

{

head = temp;

tail = temp;

}

else

{

head->prev = temp;

temp->next = head;

head = temp;

}

};

void del\_element()//function to delete index of linked list

{

int num;

bool found = false;

node\* temp = head;

cout << "Enter the Number you want to delete \n";

cin >> num;

if (head == NULL)

{

cout << "link list is Empty\n";

}

if (head == tail)

{

tail = NULL;

}

while (temp != NULL)

{

if (num == temp->data)

{

temp = head->next;

head->next = temp->next;

found = true;

cout << "Number has been delete \n";

temp = NULL;

delete(temp);

break;

}

temp = temp->next;

}

if (found == false)

{

cout << "Number Not Found \n";

}

size--;

};

void is\_empty()//function to check whether linked list is empty or not

{

if (head != NULL)

{

cout << "double link list is Not empty \n";

}

else

{

cout << "List Empty\n";

}

};

void is\_full()

{};

void display()//function to display the linked list

{

node\* temp = head;

while (temp != NULL)

{

cout << temp->data << " ";

temp = temp->next;

}

cout << endl;

}

int length()//function to check lenght of linked list

{

node\* temp=0;

int count = 0;

while (temp != NULL)

{

count++;

temp = temp->next;

}

return count;

}

void reverse() //function to reverse linked list

{

node\* temp=0;

if (temp == NULL)

return;

while(temp->next);

cout << temp->data << "\t";

}

};

int main()

{

double\_link obj;

int n;

a: cout << "Press 1 To insert the data "<<endl;

cout << "Press 2 To delete the data " << endl;

cout << "Press 3 To check if it is empty " << endl;

cout << "Press 4 To check if it is full " << endl;

cout << "Press 5 To Display " << endl;

cout << "Press 6 To check lenght " << endl;

cout << "Press 7 To reverse the data " << endl;

cout << "Press 8 To exit" << endl;

cin >> n;

if (n == 1)

{

int x;

cout << "Enter the number of data you want to add \n";

cin >> x;

int data;

cout << "Enter Data to insert\n";

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

{

cin >> data;

obj.insert(data);

}

goto a;

}

if (n == 2)

{

obj.del\_element();

goto a;

}

if (n == 3)

{

obj.is\_empty();

goto a;

}

if (n == 4)

{

obj.is\_full();

goto a;

}

if (n == 5)

{

obj.display();

goto a;

}

if (n == 6)

{

obj.length();

goto a;

}

if (n == 7)

{

obj.reverse();

goto a;

}

if (n == 8)

{

exit;

}

if (n > 8 || n < 1)

{

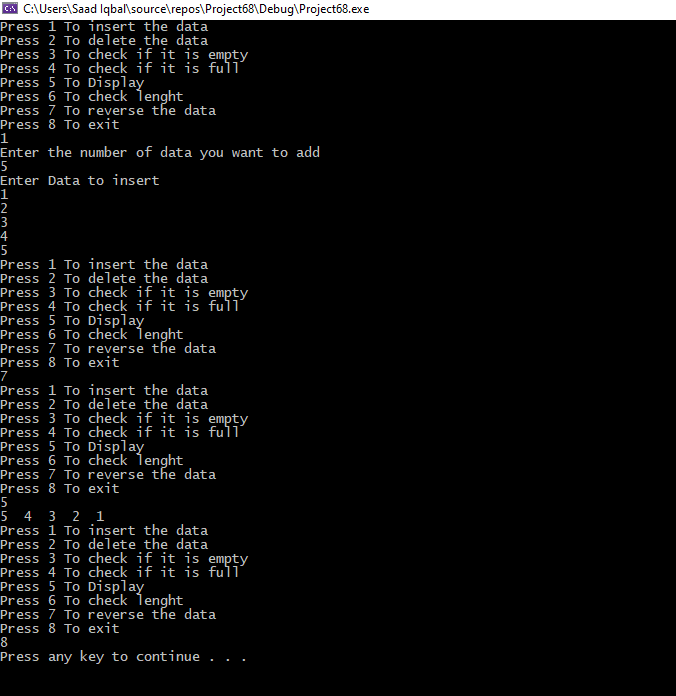
cout << "\nWrong Input\n";

goto a;

}

system("pause");

}

****

**TASK 2:**

#include<iostream>

using namespace std;

struct node

{

int num;

node\* next;

};

class list

{

private:

node\* head, \* temp;

public:

list()

{

head = NULL;

temp = NULL;

}

void input(int);

void oddData();

void display() const;

};

void list::input(int a)//function for input

{

node\* n = new node;

n->num = a;

if (head == NULL)

{

head = n;

}

else

{

temp->next = n;

}

temp = n;

n->next = NULL;

cout << a << " is added on list" << endl;

}

void list::oddData()//function for odd data

{

node\* temp2 = head, \* order = head;

while (temp2 != NULL)

{

if (temp2->num % 2 != 0)

{

int a;

a = order->num;

order->num = temp2->num;

temp2->num = a;

order = order->next;

}

temp2 = temp2->next;

}

temp2 = NULL;

order = NULL;

delete temp2;

delete order;

}

void list::display() const //function to display

{

node\* temp2 = head;

while (temp2 != NULL)

{

cout << temp2->num << "->";

temp2 = temp2->next;

}

cout << "NULL" << endl;

temp2 = NULL;

delete temp2;

}

int main()

{

int num = 0, loop = 0;

list obj;

cout << "Enter number of inputs you want to do : ";

cin >> loop;

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

{

cout << endl << "Number " << i + 1 << " : ";

cin >> num;

obj.input(num);

}

cout << endl;

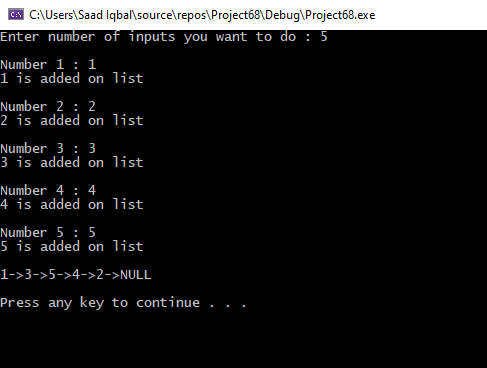
obj.oddData();

obj.display();

cout << endl;

system("pause");

}

****

**TASK 3:**

#include <iostream>

using namespace std;

void sort(int arr[], int n)

{

if (n <= 1)// Base case

{

return;

}

else

{

sort(arr, n - 1);

int last = arr[n - 1];

int j = n - 2;

for ( j=n-2;j >= 0 && arr[j] > last;j--)

{

arr[j + 1] = arr[j];

}

arr[j + 1] = last;

}

}

int main()

{

int arr[] = {2,67,54,34,6};

int n = 5;

cout << "The elements of array before sorting :"<<endl;

for (int i = 0; i < 5; i++)

{

cout << " "<<arr[i]<<endl;

}

sort(arr, n);

cout << "The elements of array after sorting :"<<endl;

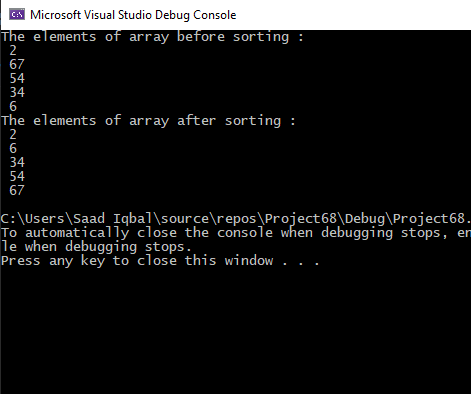
for (int i = 0; i < 5; i++)

{

cout << " " << arr[i] << endl;

}

}

****

**TASK 4:**

#include<iostream>

using namespace std;

//recursive function

int fun1(int n)

{

if (n == 1 || n == 0)//base condition

{

return n;

}

else

{

return (fun1(n - 1) + fun1(n - 2));

}

}

int main()

{

int n;

do{

int a;

cout << "\nEnter a number to which terms you want fibonacci series :";

cin >> a;

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

{

cout << fun1(i) << endl;

}

//asking user to input again

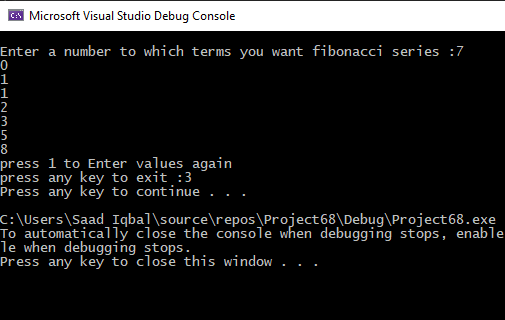
cout << "press 1 to Enter values again\npress any key to exit :";

cin >> n;

} while (n == 1);

system("pause");

}

****