**Name: Rohit Ghosh.**

**Roll Number: 21051420.**

**Date: 08/08/2022.**

**Class Assignment**

**Program 5**

**Question:**

WAP to reverse the Even Position Nodes in a Singly Linked List.

**Code:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

#include <stdint.h>

#include <stdbool.h>

typedef struct node

{

int data;

struct node \*next;

}\*NODE;

//we made the structure

NODE create(NODE start)

{

//this is the node creation function

start=NULL;

int data;

printf("\nEnter the Data. Enter -1 to terminate Input.\n");

scanf("%d",&data);

while(data!=-1)

{

NODE new\_node;

new\_node=(struct node\*)malloc(sizeof(struct node\*));

//node created for main list

new\_node->data=data;

if(start==NULL)

{

start=new\_node;

new\_node->next=NULL;

}

else

{

NODE ptr=start;

while(ptr->next!=NULL)

{

ptr=ptr->next;

}

ptr->next=new\_node;

new\_node->next=NULL;

}

printf("\nEnter the Data. Enter -1 to terminate Input.\n");

scanf("%d",&data);

}

return start;

}

//end of create function

NODE traverse(NODE start)

{

//this will make the new list

NODE start2=NULL;

NODE ptr=start;

NODE ptr2=NULL;

//we will use these two pointers further in the list

int count=1;

while(ptr!=NULL)

{

if(count%2==0) //taking out the even elements

{

NODE new\_node;

new\_node=(struct node\*)malloc(sizeof(struct node\*));

new\_node->data=ptr->data;

//new nodes are formed for new list

if(start2==NULL)

{

start2=new\_node;

new\_node->next=NULL;

}

else

{

new\_node->next=start2;

start2=new\_node;

}

//creating a new list for even elements

ptr->data=-9999; //this is a check value for display

}

ptr=ptr->next;

count++;

}

return start2;

}

//end of traverse function

void display(NODE start,NODE start2)

{

//this is the display method

int count=1;

NODE ptr=start;

NODE ptr2=start2;

printf("\n\nThe List: \n");

while(ptr!=NULL)

{

if(ptr->data!=-9999)

{

printf("%d. %d\n",count,ptr->data);

}

else

{

printf("%d. %d\n",count,ptr2->data);

ptr2=ptr2->next;

}

ptr=ptr->next;

count++;

}

}

//end of display method

int main()

{

//main execution function

NODE start=create(start);

NODE start2=traverse(start);

//all operations are now done

display(start,start2); //display method is called now

return 0;

}

//end of main method

**Output:**



