PROBLEM 1; ATTENDANCE MANAGEMENT.

QUESTION NO;1 assume that the class room has a different numbers of student both the boys and girls are presented at the total strength of 26.we mention the students with the characters in alphabets.the problem is that the students are not in the order they sitted with collapse,we need to identify that students in the attendance and classroom is same ,here the sample input and output.

INPUT ; tamil limat(CHAR)

OUTPUT; anagram(STRING)

ANSWER;

#include <stdio.h>

#include <stdlib.h>

struct Node

{

char data;

struct Node\* next;

};

struct Node\* insert(struct Node\* head, char data)

{

struct Node\* newNode = (struct Node\*)malloc(sizeof(struct Node));

newNode->data = data;

newNode->next = NULL;

if (head == NULL)

{

return newNode;

}

else

{

struct Node\* current = head;

while (current->next != NULL)

{

current = current->next;

}

current->next = newNode;

return head;

}

}

struct Node\* delete(struct Node\* head, char data)

{

if (head == NULL)

{

return head;

}

if (head->data == data)

{

struct Node\* temp = head;

head = head->next;

free(temp);

return head;

}

struct Node\* current = head;

while (current->next != NULL)

{

if (current->next->data == data)

{

struct Node\* temp = current->next;

current->next = current->next->next;

free(temp);

return head;

}

current = current->next;

}

return head;

}

int areAnagrams(const char\* str1, const char\* str2)

{

struct Node\* list1 = NULL;

struct Node\* list2 = NULL;

for (int i = 0; str1[i] != '\0'; i++)

{

list1 = insert(list1, str1[i]);

}

for (int i = 0; str2[i] != '\0'; i++)

{

list2 = insert(list2, str2[i]);

}

while (list1 != NULL)

{

if (list1->data != ' ')

{

list2 = delete(list2, list1->data);

}

list1 = list1->next;

}

return (list2 == NULL);

}

int main()

{

const char\* str1 = "tamil";

const char\* str2 = "limat";

if (areAnagrams(str1, str2))

{

printf("ANAGRAM.\n");

}

else

{

printf("NOT ANAGRAM.\n");

}

return 0;

}

PROBLEM 2; FACTOR MANAGEMENT.

QUESTION ; assume that there is a school .in the school the maths teacher ask the student to make the code for question,the staff’s question is there is a set of number that are arranged as descending order and one integer is there,we need to identify the third factor of the given integer in reverse order. (SINGLE LINKED LIST)

INPUT ; 5 (SIZE)INT

10 20 30 40 50 (ELEMENTS)INT

5 INT

OUTPUT ; 30 INT

ANSWER ;

#include <stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node\* next;

};

struct node\* top=NULL;

void create(int data)

{

struct node\* newnode = malloc(sizeof(struct node));

newnode->data=data;

newnode->next=NULL;

if(top==NULL)

{

top = newnode;

}

else

{

newnode->next=top;

top=newnode;

}

}

void display()

{

if(top==NULL)

{

printf("Empty stack");

}

struct node \*temp=top;

while(temp!=NULL)

{

printf("%d ",temp->data);

temp=temp->next;

}

}

void fact(int n)

{

int count=0;

struct node\* temp=top;

while(temp!=NULL)

{

if((temp->data)%n==0)

{

count++;

}

if(count==3)

{

printf("%d ",temp->data);

}

temp=temp->next;

}

}

int main()

{

int n,s;

printf("enter the no of stack elements...");

scanf("%d",&n);

for(int i=1;i<=n;i++)

{

scanf("%d",&s);

create(s);

}

printf("the stack elements are : ");

display();

printf("\n3rd Factor from last : ");

fact(5);

}

PROBLEM 3; SEATING ARRANGMENT BASED ON RANKING

QUESTION ; there is a school top freshers matriculation school in America . this school is completely automated.they asked the terv software solutions for software they task is assigned to mr hari and the task is to arrange the students in ascending order and give the ouput in short time for giving the seats to the students based on ranking. (SINGLE LINKED LIST)

INPUT ; 5 (SIZE) (int)

6 5 4 3 2 (int) (MAKE SORT BEFORE CREATING LL)

OUTPUT; 2 3 4 5 6 (int)

ANSWER ;

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

};

struct node \*head=NULL;

struct node \*tail=NULL;

void create(int m)

{

struct node \*newnode=(struct node\*)malloc(sizeof(struct node));

newnode->data=m;

newnode->next=NULL;

if(head==NULL)

{

head=tail=newnode;

}

else

{

tail->next=newnode;

tail=tail->next;

}

}

void display()

{

struct node \*temp=head;

if(head==NULL)

{

printf(" list is empty");

}

else

{

while(temp!=NULL)

{

printf("%d ",temp->data);

temp=temp->next;

}

}

}

int arr[50];

int main()

{

int n;

printf("enter the no of elements..");

scanf("%d",&n);

printf("enter the elements..");

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

{

scanf("%d",&arr[i]);

}

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

{

for(int j=i+1;j<n;j++)

{

if(arr[i]>arr[j])

{

int temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

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

{

create(arr[i]);

}

display();

}

PROBLEM 4; SCORE MANAGEMENT

QUESTION ; In the corporate company they allocate one day for relaxing at the day they make some games for employees they divide the employees into two teams and conduct the games for individual persons.the task is to find the winning team by calculating the individual score and find the total score of team and find the difference between two teams .finally we announce the winner with the difference between the both team(TWO SINGLE LINKED LIST)

INPUT ; 1 2 3 4 team 1 (int) 10

6 7 8 4 team2 (int) 25

OUTPUT ; team 2 lead with 15 (int) (DIFFERENCE BTW TO SEGMENTS)

ANSWER ;

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

struct node

{

int data;

struct node \*next;

};

struct node \*head1=NULL;

struct node \*head2=NULL;

struct node \*tail1=NULL;

struct node \*tail2=NULL;

void create1(int a)

{

struct node\* newnode=(struct node\*)malloc(sizeof(struct node));

newnode->data=a;

newnode->next=NULL;

if(head1==NULL)

{

head1=tail1=newnode;

}

else

{

tail1->next=newnode;

tail1=tail1->next;

}

}

void create2(int a)

{

struct node\* newnode=(struct node\*)malloc(sizeof(struct node));

newnode->data=a;

newnode->next=NULL;

if(head2==NULL)

{

head2=tail2=newnode;

}

else

{

tail2->next=newnode;

tail2=tail2->next;

}

}

void display1()

{

struct node \*temp=head1;

if(head1==NULL)

{

printf("Empty");

return;

}

printf("list2 elements are...");

while(temp!=NULL)

{

printf("%d ",temp->data);

temp=temp->next;

}

}

void display2()

{

struct node \*temp=head2;

if(head2==NULL)

{

printf("Empty");

return;

}

printf("list1 elements are...");

while(temp!=NULL)

{

printf("%d ",temp->data);

temp=temp->next;

}

}

int sum1()

{

int c=0;

struct node \*temp=head1;

if(head1==NULL)

{

printf("Empty");

return 0;

}

else

{

while(temp!=NULL)

{

c=c+temp->data;

temp=temp->next;

}

}

return c;

}

int sum2()

{

int c=0;

struct node \*temp=head2;

if(head2==NULL)

{

printf("Empty");

return 0;

}

else

{

while(temp!=NULL)

{

c=c+temp->data;

temp=temp->next;

}

}

return c;

}

int main()

{

create1(1);

create1(2);

create1(3);

create1(4);

create2(6);

create2(7);

create2(8);

create2(4);

display1();

printf("\n");

int S1=sum1();

display2();

printf("\n");

int S2=sum2();

printf("%d",abs(S1-S2));

}

PROBLEM 5; IQ MANAGEMENT

QUESTION ; there is a task that used to identify the iq for the human.the doctor can write can show the different characters we need to convert the uppercase to lower case within the short time ( SINGLE LINKED LIST)

INPUT ; CaR (char)

OUTPUT ; cAr (char) (CASE CONVERSION)

ANSWER ;

#include <stdio.h>

#include <stdlib.h>

struct Node

{

char data;

struct Node\* next;

};

struct Node\* head = NULL;

void create(int n)

{

struct Node\* temp = NULL;

struct Node\* newnode = NULL;

char data;

for (int i = 1; i <= n; i++)

{

newnode = (struct Node\*)malloc(sizeof(struct Node));

scanf(" %c", &data);

newnode->data = data;

newnode->next = NULL;

if (temp == NULL)

{

head = newnode;

temp = newnode;

} else

{

temp->next = newnode;

temp = newnode;

}

}

}

void display()

{

struct Node\* temp = head;

if (temp == NULL)

{

printf("Memory wasn't created.\n");

} else

{

while (temp != NULL)

{

char c = temp->data;

int n = (int)c;

int ans = n ^ 32;

printf("%c\n", (char)ans);

temp = temp->next;

}

}

}

int main()

{

int n;

printf("number of nodes: ");

scanf("%d", &n);

create(n);

display();

return 0;

}