///Md.Inzamam-Ul-Haque

//Roll:85

//stack implementation

#include<bits/stdc++.h>

using namespace std;

int sz=0;

struct Node

{

int Value;

struct Node \*Next;

};

struct Node \*Head, \*LL, \*Temp,\*Tail,\*pot,\*pot\_n,\*love;//Globally Declared node type struct pointer

void Create\_A\_Node(int V)

{

Temp = (struct Node \*)malloc(sizeof(struct Node));//Dynamic memory allocation

if(!Temp)

{

printf("Something Wrong with Node Creation\n");

exit(0);

}

else

{

Temp->Value = V;//assign value

Temp->Next = NULL;//assign NULL

}

}

void Free\_A\_Node(struct Node \*A\_Node)

{

free(A\_Node);//memory free

}

void push()

{

int x;

printf("\nEnter a number: ");

cin>>x;

LL=Head;

if(sz==0)

{

Create\_A\_Node(x);

Head=Temp;

}

else

{

while(LL->Next) // LL will traverse till the end of the Linked List or the Node that has Null in it's next

{

LL = LL->Next;

}

Create\_A\_Node(x); // Calling this function with V and the new Node will be created as Temp;

LL->Next = Temp;

}

sz++;

}

void pop()

{

LL = Head;

if(sz==0)

{

puts("No element in stack");

}

else if(sz==1)

{

Free\_A\_Node(Head);

Free\_A\_Node(LL);

sz--;

}

else

{

while(1)

{

if(LL->Next->Next==NULL)

{

LL->Next=NULL;

break;

}

else

LL=LL->Next;

}

sz--;

}

}

void top()

{

LL=Head;

if(sz==0)

{

puts("No element in Stack");

}

else

{

int c;

c=sz;

while(--c)

{

LL=LL->Next;

}

printf("%d\n",LL->Value);

}

}

void display()

{

LL = Head;

if(sz==0)

return;

while(LL!=NULL) // Or you can write while(LL)

{

printf("%d\n",LL->Value);

LL= LL->Next;

}

}

void menu()

{

puts("For push press 1");

puts("For pop press 2");

puts("For Top press 3");

puts("For Display press 4");

puts("For Exit press 0");

}

int main()

{

int n;

while(1)

{

menu();

printf("\nEnter option No.: ");

cin>>n;

if(n==0)

break;

else if(n==1)

push();

else if(n==2)

pop();

else if(n==4)

display();

else if(n==3)

top();

}

return 0;

}

//queue implementation

#include<bits/stdc++.h>

using namespace std;

int sz=0;

struct Node

{

int Value;

struct Node \*Next;

};

struct Node \*Head, \*LL, \*Temp,\*Tail,\*pot,\*pot\_n,\*love;//Globally Declared node type struct pointer

void Create\_A\_Node(int V)

{

Temp = (struct Node \*)malloc(sizeof(struct Node));//Dynamic memory allocation

if(!Temp)

{

printf("Something Wrong with Node Creation\n");

exit(0);

}

else

{

Temp->Value = V;//assign value

Temp->Next = NULL;//assign NULL

}

}

void Free\_A\_Node(struct Node \*A\_Node)

{

free(A\_Node);//memory free

}

void push()

{

int x;

printf("\nEnter a number: ");

cin>>x;

LL=Head;

if(sz==0)

{

Create\_A\_Node(x);

Head=Temp;

}

else

{

while(LL->Next) // LL will traverse till the end of the Linked List or the Node that has Null in it's next

{

LL = LL->Next;

}

Create\_A\_Node(x); // Calling this function with V and the new Node will be created as Temp;

LL->Next = Temp;

}

sz++;

}

void pop()

{

LL = Head;

if(sz==0)

{

puts("No element in queue");

}

else if(sz==1)

{

Free\_A\_Node(Head);

Free\_A\_Node(LL);

sz--;

}

else

{

LL=Head;

Head=LL->Next;

sz--;

}

}

void top()

{

LL=Head;

if(sz==0)

{

puts("No element in Stack");

}

else

{

printf("%d\n",LL->Value);

}

}

void display()

{

LL = Head;

if(sz==0)

return;

while(LL!=NULL) // Or you can write while(LL)

{

printf("%d\n",LL->Value);

LL= LL->Next;

}

}

void menu()

{

puts("For push press 1");

puts("For pop press 2");

puts("For Front press 3");

puts("For Display press 4");

puts("For Exit press 0");

}

int main()

{

int n;

while(1)

{

menu();

printf("\nEnter option No.: ");

cin>>n;

if(n==0)

break;

else if(n==1)

push();

else if(n==2)

pop();

else if(n==4)

display();

else if(n==3)

top();

}

return 0;

}

#include<bits/stdc++.h>

using namespace std;

int main()

{

int i,j,k,n;

puts("Enter a number:");

cin>>n;

double low =0.0,high,mid;

high = n;

while(high-low>=0.0000001)

{

mid=(double)(high+low)/2.0;

// cout<<mid<<" "<<low<<" "<<high<<endl;

if(mid\*mid==n)

{

break;

}

else if((mid\*mid)<n)

{

low=mid;

}

else

{

high = mid;

}

}

printf("%lf\n",mid);

return 0;

}

#include<bits/stdc++.h>

using namespace std;

int main()

{

int i,k,n;

puts("Enter two number k & n :");

cin>>k>>n;

double low =0.0,high,mid,j;

high = n;

while(high-low>=0.0000000001)

{

mid=(double)(high+low)/2.0;

//cout<<mid<<" "<<low<<" "<<high<<endl;

j=pow(mid,k);

if(j==n)

{

break;

}

else if(j<n)

{

low=mid;

}

else

{

high = mid;

}

}

printf("%lf\n",mid);

return 0;

}

#include<bits/stdc++.h>

using namespace std;

int main()

{

int i,k;

double u,v,s,temp;

puts("Enter u & v & s:");

cin>>u>>v>>s;

if(u<v)

{

temp=u;

u=v;

v=temp;

}

double low =0.0,high,mid,m,n;

high = s;

while(high-low>=0.0000000001)

{

mid=(double)(high+low)/2.0;

//cout<<mid<<" "<<low<<" "<<high<<endl;

m=(double)mid/v;

n=(double)(s-mid)/u;

if(m==n)

{

break;

}

else if(m<n)

{

low=mid;

}

else

{

high = mid;

}

}

printf("%lf\n",mid);

main();

return 0;

}