1)СОРТУВАННЯ СПИЧКІВ ШЛЯХОМ ЗЛИТТЯ

#include <iostream>

using namespace std;

void merge(int \*mas, int l, int m, int r)

{

int n1 = m - l + 1;

int \*a = new int [n1];

int n2 = r - m;

int \*b = new int[n2];

for (int i = 0; i < n1; i++)

a[i] = mas[l + i];

for (int i = 0; i < n2; i++)

b[i] = mas[m + 1 + i];

int i1 = 0;

int i2 = 0;

int i;

for (i = l; i < l + n1 + n2 && i1<n1 && i2<n2; i++)

{

if (a[i1] > b[i2])

{

mas[i] = b[i2];

i2++;

}

else

{

mas[i] = a[i1];

i1++;

}

}

for (i1; i1 < n1; i1++, i++)

mas[i] = a[i1];

for (i2; i2 < n2; i2++, i++)

mas[i] = b[i2];

}

void mergesort(int \*mas, int l, int r)

{

if (l < r)

{

int m = (l + r) / 2;

mergesort(mas, l, m);

mergesort(mas, m + 1, r);

merge(mas, l, m, r);

}

}

int main()

{

int \*mas;

int n;

cout << "Vvedit' chyslo elementiv masyvu" << endl;

cin >> n;

mas = new int[n];

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

{

cout << "Vvedit' element masyvu" << endl;

cin >> mas[i];

}

mergesort(mas, 0, n - 1);

cout << "Masyv:" << endl;

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

{

cout << mas[i] <<" ";

}

return 0;

}

2)ШВИДКЕ СОРТУВАННЯ

#include <iostream>

using namespace std;

void quicksort(int \*mas, int l, int r)

{

if (l<r)

{

int m = (l + r) / 2;

int v = mas[m];

int i = l;

int j = r;

do

{

while (mas[i] < v)

i++;

while (mas[j]>v)

j--;

if (i < j)

{

int temp = mas[i];

mas[i] = mas[j];

mas[j] = temp;

cout << "Masyv:" << endl;

for (int i = 0; i < 6; i++)

{

cout << mas[i] << " ";

}

}

//i++;

//j--;

} while (i < j);

quicksort(mas, l, j-1);

quicksort(mas, i+1, r);

}

}

int main()

{

int \*mas;

int n;

cout << "Vvedit' chyslo elementiv masyvu" << endl;

cin >> n;

mas = new int[n];

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

{

cout << "Vvedit' element masyvu" << endl;

cin >> mas[i];

}

quicksort(mas, 0, n-1);

cout << "Masyv:" << endl;

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

{

cout << mas[i] <<" ";

}

return 0;

}

3)СОРТУВАННЯ ЗА ДОПОМОГОЮ ВИБОРУ МІНІМАЛЬНОГО ТА МАКСЕМАЛЬНОГО ЕЛЕМЕНТА

#include <iostream>

using namespace std;

void selectsort(int \*mas, int l, int r)

{

int min ;

int max ;

int imin;

int imax;

while (l < r)

{

min = mas[l];

max = mas[r];

imin = l;

imax = r;

for (int i = l + 1; i <= r; i++)

{

if (mas[i] < min)

{

min = mas[i];

imin = i;

}

}

int temp = mas[l];

mas[l] = mas[imin];

mas[imin] = temp;

for (int i = r-1; i >= l; i--)

{

if (mas[i]>max)

{

max = mas[i];

imax = i;

}

}

temp = mas[r];

mas[r] = mas[imax];

mas[imax] = temp;

l++;

r--;

}

}

int main()

{

int \*mas;

int n;

cout << "Vvedit' chyslo elementiv masyvu" << endl;

cin >> n;

mas = new int[n];

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

{

cout << "Vvedit' element masyvu" << endl;

cin >> mas[i];

}

selectsort(mas, 0, n - 1);

cout << "Masyv:" << endl;

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

{

cout << mas[i] <<" ";

}

return 0;

}

4)МАВПЯЧЕ СОРТУВАННЯ

#include <iostream>

using namespace std;

void masyv()

{

int \*mas;

int n;

cout<<"Vvedit' kilkist' elementiv"<<endl;

cin>>n;

mas= new int[n];

int i;

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

{

cout<<"Vvedit' element masyvu"<<endl;

cin>>mas[i];

}

cout<<"Vyvodymo elementy masyvu"<<endl;

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

{

cout<<mas[i]<<" ";

}

// мавпяче сортування

int i1;

int i2;

int temp;

int flag;

do

{

flag=0;

for(i=0;i<n-1;i++)

{

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

{

flag=1;

break;

}

}

if(flag==0)

break;

i1=rand()%n;

i2=rand()%n;

temp=mas[i1];

mas[i1]=mas[i2];

mas[i2]=temp;

}

while(1);

cout<<endl<<"Vyvodymo elementy masyvu"<<endl;

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

{

cout<<mas[i]<<" ";

}

}

int main()

{

//масив

masyv();

return 0;

}

5)СТЕК, ЧЕРГА, ДВОНАПРЯМЛЕНИЙ СПИСОК

#include <iostream>

using namespace std;

struct node

{

int elem;

struct node \*next;

struct node \*prev; // для 2-напрямленого списку

};

void stack()

{

struct node \*top=NULL, \*curr=NULL;

int x;

do

{

cout<<"Vvedit element steku"<<endl;

cin>> x;

if(x==0)

break;

curr= new node;

curr->elem = x;

curr->next=top; // саме стек

top=curr; // саме стек

}

while(1);

cout<<"Vyvodymo stek"<<endl;

curr=top;

while(curr!=NULL)

{

cout<<curr->elem<<" ";

curr=curr->next;

}

}

void cherga()

{

struct node \*head=NULL, \*tail=NULL, \*curr=NULL;

int x;

do

{

cout<<"Vvedit' element chergy"<<endl;

cin>>x;

if(x==0)

break;

curr= new node;

curr->elem=x;

curr->next=NULL;

if(tail==NULL)

{

head=curr;// для черги

tail=curr;// для черги

}

else

{

tail->next=curr;// для черги

tail=curr;// для черги

}

}while(1);

cout<<"Vyvodymo chergu"<<endl;

curr=head;

while(curr!=NULL)

{

cout<<curr->elem<<" ";

curr=curr->next;

}

}

void spysok()

{

struct node \*head=NULL, \*tail=NULL, \*curr=NULL;

int x;

do

{

cout<<"Vvedit' element spysku"<<endl;

cin>>x;

if(x==0)

break;

curr= new node;

curr->elem=x;

curr->next=NULL;

curr->prev=NULL;

if(tail==NULL)

{

head=curr;// для черги

tail=curr;// для черги

}

else

{

tail->next=curr;// для черги

curr->prev=tail;// для 2-напрямленого списку (схоже як для стеку: curr->next=top; )

tail=curr;// для черги

}

}while(1);

cout<<"Vyvodymo spysok z golovy"<<endl;

curr=head;

while(curr!=NULL)

{

cout<<curr->elem<<" ";

curr=curr->next;

}

cout<<endl<<"Vyvodymo spysok z hvosta"<<endl;

curr=tail;

while(curr!=NULL)

{

cout<<curr->elem<<" ";

curr=curr->prev;

}

}

int main()

{

// стек

//stack();

// черга

//cherga();

// двонапрямлений список (черга+стек)

//spysok();

return 0;

}

6)СОРТУВАННЯ ЧОВНИКОМ

#include <iostream>

using namespace std;

int main()

{

int n=4;

int temp=0;

int mas[4][3]={ {12,11,10},{9,8,7}, {6,5,4},{3,2,1}};

int mas1[4];

for(int i=0;i<4;++i)

{

mas1[i]=0;

}

for(int i=0;i<4;++i)

{

for(int j=0;j<3;++j)

{

mas1[i]+=mas[i][j];

}

}

int l=1, r=n-1,k=n-1;

while(l<=r)

{

for(int j=r;j>=l;--j)

{

if(mas1[j-1]>mas1[j])

{

temp=mas1[j];

mas1[j]=mas1[j-1];

mas1[j-1]=temp;

k=j;

for(int j1=0; j1<3; j1++)

{

temp=mas[j][j1];

mas[j][j1]=mas[j-1][j1];

mas[j-1][j1]=temp;

}

}

}

l=k+1;

for(int j=l;j<=r;++j)

{

if(mas1[j-1]>mas1[j])

{

temp=mas1[j];

mas1[j]=mas1[j-1];

mas1[j-1]=temp;

k=j;

for(int j1=0; j1<3; j1++)

{

temp=mas[j][j1];

mas[j][j1]=mas[j-1][j1];

mas[j-1][j1]=temp;

}

}

}

r=k-1;

}

return 0;

}

7)Задана послідовність з N різних натуральних чисел,

представлених списком. Надрукувати в оберненому

порядку всі числа між найбільшим та найменшим

елементами послідовності.

#include <iostream>

using namespace std;

int main()

{

struct stack

{

int data;

struct stack \*next;

};

struct stack \*st1 = NULL, \*newel;

int n;

cout << "n = "; cin >> n;

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

{

newel = new stack;

cin >> newel->data;

newel->next = st1;

st1 = newel;

}

int i1, i2, max, min;

newel = st1;

max = st1->data;

i1 = 1;

for (int i = 2; i <= n; ++i)

{

newel = newel->next;

if (newel->data>max)

{

max = newel->data;

i1 = i;

}

}

newel = st1;

min = st1->data;

i2 = 1;

for (int i = 2; i <= n; ++i)

{

newel = newel->next;

if (newel->data<min)

{

min = newel->data;

i2 = i;

}

}

cout <<endl<< i1 << endl << i2 << endl;

newel = st1;

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

{

if (i >= i1&&i <= i2 || i <= i1&&i >= i2)

cout << newel->data << " ";

newel = newel->next;

}

return 0;

}

8)Задано натуральне число. Створити стек з його цифр.

#include <iostream>

#include <math.h>

using namespace std;

int main()

{

int n, rez = 0, a = 0;

cout << "n = "; cin >> n;

struct stack

{

int data;

struct stack \*next;

};

struct stack \*st1 = NULL, \*newel;

/\*do

{

newel = new stack;

newel->data = n % 10;

newel->next=st1;

st1 = newel;

n /= 10;

rez++;

//cout << newel->data << " ";

}

while (n != 0);

newel=st1;

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

{

cout << newel->data << " ";

newel=newel->next;

}\*/

int b = n;

/\*do

{

b/=10;

rez++;

}

while(b!=0);\*/

for (int i = 0; i<100000; i++)

rez = log10((double)n) + 1;

do

{

a = pow(10.0, rez - 1);

newel = new stack;

newel->data = n / a;

newel->next = st1;

st1 = newel;

n %= a;

rez--;

} while (rez>0);

newel = st1;

for (int i = 1; newel != 0; ++i)

{

cout << newel->data << " ";

newel = newel->next;

}

return 0;

}

9)Задано ціле число N. Використовуючи метод “решето Ератосфена” надрукувати в зростаючому порядку всі прості числа з діапазону N..2N

#include <iostream>

using namespace std;

int main()

{

struct stack

{

int data;

struct stack \*next;

};

struct stack \*st1, \*newel, \*oldel;

int n;

cout << "n = "; cin >> n;

for (int i = n; i <= 2 \* n; ++i)

{

newel = new stack;

if (i != n)

oldel->next = newel;

newel->data = i;

newel->next = NULL;

if (i == n)

st1 = newel;

oldel = newel;

}

/\*newel=st1;

for(int i=n;i<=2\*n;++i)

{

cout<<newel->data<<" ";

newel=newel->next;

}\*/

int d = 2;

do

{

newel = st1;

oldel = st1;

for (int i = n; newel->data <= 2 \* n; ++i)

{

if (newel->data%d == 0 && newel->data != 2)

{

if (newel != st1)

{

oldel->next = newel->next;

delete newel;

newel = oldel->next;

}

else

{

st1 = newel->next;

delete newel;

newel = st1;

}

}

oldel = newel;

if (newel != NULL && newel->next != NULL)

newel = newel->next;

else

break;

}

d++;

} while (d<n);

int j = 0;

newel = st1;

for (int i = n; newel->data <= 2 \* n; ++i)

{

cout<<newel->data<<" ";

++j;

if (newel->next != NULL)

newel = newel->next;

else

break;

}

//cout << "j = " << j;

return 0;

}

10)N дітей розміщені по колу. Починаючи зпершого, видаляють з кола кожного к- ого , змикаючи коло після кожного видалення. Визначити порядок видалення дітей з кола.

#include <iostream>

using namespace std;

int main()

{

int n;

cout<<"n = ";cin>>n;

struct stack

{

int data;

struct stack \*next;

};

struct stack \*st1=NULL,\*newel, \*oldel;

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

{

newel=new stack;

if(i!=1)

oldel->next=newel;

newel->data=i;

if(i==1)

st1=newel;

newel->next=st1;

oldel=newel;

}

/\*newel=st1;

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

{

cout<<newel->data<<" ";

newel=newel->next;

}\*/

int k;

cout<<"k = ";cin>>k;

newel=st1;

do

{

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

{

oldel=newel;

newel=newel->next;

}

cout<<newel->data<<" ";

oldel->next=newel->next;

delete newel;

newel=oldel->next;

n--;

}

while(n!=0);

return 0;

}

11)Задано натуральне число. Створити стек з його цифр.

#include <iostream>

using namespace std;

int main()

{

int n, rez = 0;

cout << "n = "; cin >> n;

struct stack

{

int data;

struct stack \*next;

};

struct stack \*st1, \*newel;

do

{

newel = new stack;

newel->data = n % 10;

st1 = newel;

newel->next = NULL;

n /= 10;

cout << newel->data << " ";

} while (n != 0);

return 0;

}

12)Число вводиться у вигляді стрічки символів. Представити його у вигляді стеку та підрахувати суму його цифр.

#include <iostream>

#include <stdio.h>

using namespace std;

int main()

{

int i=0, sum=0;

struct stack

{

char data;

struct stack \*next;

};

struct stack \*st1=NULL, \*newel;

char mas[100];

cin>>mas;

while(mas[i]!='\0')

{

newel=new stack;

newel->data= mas[i];

i++;

newel->next=st1;

st1=newel;

}

fflush(stdin);

newel=st1;

do

{

sum+=newel->data-'0';

cout<<newel->data;

newel=newel->next;

}

while(newel!=NULL);

cout<<endl<<"sum of digits = "<<sum<<endl;

return 0;

}