C++

**# include <bits/stdc++.h> hammasi uchun kutubhona**

//setw( 13 ) 13 ta boshliq tashlash uchun ishlatiladi

// -> bularni diqqat bilan o'qib chiqin

// manli fayllar

// !!! ofstream yoz("2.txt",ios::app) matnni ochirmasdan davomidan yozish

// eng.assign(s,0,p-1); eng ozgaruvchisiga 0 inci elementdan boshlab p-1 ta element nusxalandi

// fromfile.read(buffer, length); bolkka malumotlani oqish

// delete[] buffer; hotirani boshatish

// buffer = new char [length]; hotira ajratish

// fromfile.seekg(0, ios::beg); // fayl boshiga otish

// fromfile.seekg(0, ios::end); // fayl ohririga otish

// length=fromfile.tellg(); fayl hajmi aniqlanadi

// ofstream tofile(yangi, ios::binary); yozish uchun binar fayl ochish

// long tellp() yozsih oqimnig kelgan jooyini aniqlash

// ios::beg oqimning boshlannishi

// ios::cur oqimning joriy holati

// ios::end oqim oxiri

// oqish.eof() satr ohiri

// while(!oqish.eof()) satr ohiriga yetmagan paytda

// if(!oqish.is\_open()) agar fayl ochilmasa deganni bildiradi

// yozish.open("matn01.txt"); shu matnga yozish

// yozish.close(); yozish uchun ochilgan faylni yopish

// oqish.close(); oqish uchun ochilgan faylni yopish

// ofstream ios::out

// ifstream ios::in

// fstream ios::in|ios::out

// har bir sinf uchun mode parametrinig odatiy qiymatlari mavjud

// ios::in -> faqat malumot o'qish uchun

// ios::out -> faqat malumot yozish uchun ishlatiladi

// ios::ate -> faylni ochishda fayl korsatkichini fayl ohiriga qoyish uchun ishlatiladi

// ios::app -> fayl ohiriga malumot yozish uchun ishlatiladi

// ios::trunc -> bor bo'lgan faylni ustidan yangi fayl yozish

// ios::binary -> binar holda malumotlarni almshtirish uchun ishlatiladi

// #include <fstream> kutubhona qoshiladi

// fstream fayldan malumot o'qish va yozish uchun ishlatiladi fstream yoz\_oqi;

// ifstream fayldan malumot o'qish uchun ishlatiladi ifstream oqi;

// ofstream faylga malumot yozish uchun ishlatiladi ofstream yoz;

// binar fayllar

// fseek(f,-sizeof(int),SEEK\_CUR); cursorni siljitish

// fwrite(&k,sizeof(int),1,f); yozish

// fseek(f, -sizeof(int), SEEK\_CUR); chapga surish

// fseek(f,ftell(f),SEEK\_SET); siljitish

// fseek(f,ftell(f),SEEK\_SET); fayl korsatkichini ornatib qoyish

// fseek(f, 0, SEEK\_END); fayl ohiriga otish

// SEEK\_SET fayl boshiga nisbatan siljish

// SEEK\_CUR fayl joriyga nisbatan siljish

// SEEK\_END fayl ihiriga nisbatan siljish

// fseek fayl ni siljitish uchun ishlatiladi

// ftell(f); bu funksiya orqali fayl hajmi qancha ekanlii koriladi

// rewind(f) -> fayl korsatkichini fayl bshiga ornatadi

// feof(f) fayl ohiri

// fread(&k,sizeof(k),1,f); faylni oqish

// fwrite(&k,sizeof(k),1,f); yozish

// fclose(f); faylniyopish uchun ishlatiladi

// binar fayllar uchun #include <stdio.h> kutubhona ishlatiladi

// FILE \*f; elon qilish

// f=fopen("1.dat","wb"); yozish uchun fayl ochish

// f=fopen("1.dat","wb+"); yozish va oqish uchun fayl ochish

// rekursiya

// oz ozini chaqiruvchi funksiyalar

// if(k==0||k==1)

// return 1;

// return k\*fact(k-1);

// string!!!

// s1.replace(p,s2.length(),s3); s1 satrga p-incidan boshlab s2 ni uzunlgicha s3 satrni qoyish

// index=s.find(c,index+s1.length()); yana qidirish

// s.replace(index,c.length(),s1); s satrning c satrnini orniga s1 satrni qoyish

// cout<<"s.rfind('a')"<<s.rfind("a")<<endl; // s satrdan 'a' ohirgi uchragan belgini qidirish

// cout<<"s.find('a')"<<s.find("a")<<endl; // s satrdan 'a' 1-uchragan belgini qidirish

// cout<<"s.find(c)="<<s.find(c)<<endl; // 9-belgidan boshlab uchradi

// s="guloy"; c=s.substr(3.2)+s.substr(0.3); bolda s="oygul" chiqadi

// substr() satrlarni qoshish uchun ishlatiladi

// cout<<"c.compare(2,7,s,9,7)="<<c.compare(2,7,s,9,7)<<endl; // 0 bular teng

// cout<<"s.compare(c)="<<s.compare(c)<<endl; // -1 s c dan kichik bogani uchun -1 qaytaradi

// s.compare(c); satrlarni solishtirish

// s.replace(0,0,c,0,17); s satrni 0- elementidan boshlab 0 ta elementi

// c satr= 0 -elementidan 17 ta element qoshilyapti

// s.erase(25); 25 - belgidan boshllab tozalanyapti

// s.swap(c); // 2 ta satrni tola almashtirish

// s.replace(17,5,c); 17-belgidan boshlab 5 ta beelgi o'rniga c satrini qoshish

// s.insert(0,p,13); char toiifasidagi satrdan 13 ta belgi qoshish

// s.insert(21,c,8,10); c satridan 8-belgidan boshlab 10 ta belgi qoshish

// s.insert(17,"aziz\_"); 17-belgidan boshlab yangi satr qoshish

// s.erase() bolsa satr hammasi o'chiriladi s.clear() vazifa bir hil

// s.erase(16,9); bu funksiya 16 - o'rundan boshlab 9 ta belgi o'chiradi agar 9 bolmasa ohirigacha ochirdi

// string toifasidagi satrni char toifasiga ogirish uchun c\_str() yoki data() ishlatiladi datada satr ohiri belgisi qoyilmidi '\0'

// strcpy(c,s.c\_str()); string toifasidagi satrni char toifasiga ogirish

// s.assign(c,strlen(c)); char toifasidagi c dan string toifasiga otish

// s.assign(s2,a,b) da malumotlar korsatiladi va nusha olinsa s.append(s2) da davomiga qoshiladi

// s1="GULBAHOR";

// s2=GUL bolgani uchhun

// s2.append(s3); // s2="GULBAHOR" bunda toliq usha olinadi

// s2.assign(s1,0,3); // s2="GUL" s1 satrni 0 inchi elementidan boshlab 3 ta belgi qirqib olindi

// void clear(); satrni tozalash bu s=""; belgi bilan bir hil

// bool empty() const -> satr bosh ekanliini tekshiradi agar bosh bolsa true qaytaradi

// satr uzunligini ozgartirish uchun ressize funksiyadan foydalaniladi

// void resize ( size\_t n, char c); void resize ( size\_t n);

// n=str.size(); -> satr uzunligini aniqlash

// str.resize(n+2, '+'); -> satr uzunligini 2 ta belgiga uzaytirish

// str resize(14) - > satr uzunligini ozgartirish 14 ta belgin qoladi

// cout<<s.length(); <-> cout<<s.size(); teng kuchli belgilari soni

// satrlar bilan ishlovchi asosiy funksiyalar bilan tanishib chiqamiz

// unsigned int size() const; -> satr olchamini aniqlash uchun

// unsigned int length() const; -> satr elementlar sonini aniqlash uchun

// unsigned int max\_size() const; -> satr maksimal uzunligini aniqlash uchun

// unsigned int capacity() const; -> satr egallagan xotira hajmini aniqlash uchun

// bool empty() const; -> satr boshligini aniqlash uchun

// string satrlari uchun #include <string> kutubhonasi ishlatiladi bazi compilyatorladrda <cstring.h> <bstring.h>

// getline(cin,s); string toifadagi satrni kiritish uchun ishlatiladi!!!

// #include <string.h> <- = -> yoki #include <cstring.h>

// char

// char s[s];

// gets(s) s satrni kiritish uchun ishlatiladi

// puts(s) s strni chiqarish uchun ishlatiladi

// cin.getline(s,sizeof(s)); char toifadagi satrni kiritish

// itoa(n,onlik,10); 10 likdagi sonni satrga ogirish

// n=strtol(ikkilik,&p,2); 2 likdan onliik sanoq tizimiga otish

// cout<<strupr(m)<<endl; bu kichik harflani katta qiladi

// itoa(n,on\_olti,16); 16 likga otkazish

// itoa(n,ikki,2); 2 likga otkazish

// ta1=strtol(sonlar,&pEnd,16); 16 likga otkazish

// atof -> floatga otkazadi

// atoi -> int ga otkkazadi

// agar biz toifanin alashtirmoqchi bolsek stdlib.h lutubhona qoshiladi

// double atof (const char \*c) => c satrni double toifasigga ozgartiradi

// int atoi (const char \*c) => c satrni int toifasigga ozgartiradi

// int atol (const char \*c) => c satrni long int toifasigga ozgartiradi

// int strtol (const char \*c,char \*\*endptr,int base) => c satrni long int toifasigga ozgartiradi

// double strtod (const char \*c,char \*\*endptr) => c satrni double toifasigga ozgartiradi

// char \*itoa (int n, char \*satr,int radix) => n sonini radix sanoq sistemasida satr ozgaruvchisiga ozlashtiradi

// char \*gcvt (double n, int k,char \*satr) => haqiqiy toifadagi n sonining k ta raqamini satr ozgaruvchisiga ozlashtiradi

// toupper har harfni katta qilishuchun ishlatiladi

// tolower har harfni kichik qilishuchun ishlatiladi

// belgilarni qayta ishlovchi funksiyalar

// buning uchun ctype.h kutubhonna qoshiladi

// int isdigit(int c) agar c raqam bolsa true aks holda false qaytaradi

// int isalpha(int c) agar c harp bolsa true aks holda false qaytaradi

// int isalnum(int c) agar c raqam yoki harp bolsa true aks holda false qaytaradi

// int islower(int c) agar c kichik harp bolsa true aks holda false qaytaradi

// int isupper(int c) agar c katta harp bolsa true aks holda false qaytaradi

// int istolower(int c) agar c katta harp bolsa kichik harp qaytaradi aks holda tolower argumenti ozgarish qaytaradi

// strtok(s," ,.-"); strtok funksiyasi satrni ajraataadi agar shu beligilaf kelsa satr ajraydi n taga

// s="salom"; <- yoki -> cin.getline(s,sizeof(s));

// cout<<sizeof(s) - > 20;

// couut<<strlen(s)- > 5;

// bularga string.h kutubhon qoshialdi

// strlen satr uzunligini aniqlash strlen(s)- > 5;

// strcpy, ctrncpy - satrdan nusha olish strcpy(s1, "oddiy satr"); bunda satr ohiriga satr ohiri belgisi qoyiladi '\0'

// strcat, strncat - satrlarni qoshish (ulash) strncat(s1, "dasturchilar ", 9);

// strcmp, strncmp - satrlarni solishtirish (strcmp(parol, s)!=0)

// strstr - qidirish uchun ishlatiladi (strstr(str[i], s) !=NULL)

// strtok - satrni qismlarga ajrarish strtok(NULL," ,.-");

// char satr[15];

// cin.getline(satr, 15); cin.getline(satr, sizeof(satr));

// satr ohiiri '\0' belgisi bilan tugaaedi

// c=getche(); belgi kiritish buning uchun bunda belgi chiqadi nclude <conio.h>

// for(char i='a'; i<='z'; i++) cout<<"->"<<(int)i<<endl;

// for(int i=65; i<=90; i++) cout<<i<<"->"<<(char)i<<endl;

// matritsalar

// a[100][100]; statik matritsa elon qilish.

// dinamik matritsa elon qilish \*\*a => new int \*[satrlar\_soni];

// har bir satrga takrorlash yordamida boshlangich qiymat beriladi

// for() => a[i]=new int [ustunlarr soni];

// siklni ochirish for(int i=0; i<satrlar soni; i++)

// delete [] a[i]; avvalsatrlar ochiriladi

// delete []a; keyin massiv ochiriladi.

// massivlar

// C++ da dinamik massiv hosil qilish\/

// int \*a; korsatkich elon qilamiz va a=new int[n]

// hotirani boshatish uchun delete []a;

// C stilida dinamik massiv hosil qilish

// int \*a -> va pastdan a=(int\*) malloc(n\*sizeof(int)); \/

// bunga kereyli kutubhona #include <alloc.h>

// dinamik massivdan foydalanganda dastur ohirida hotirani boshatamiz -> free(a);

// massiv elementlar sonini aniqlash cout<<sizeof(a)/sizeof(int);

// massiv elementini hotirada turgan orni ni olish &a[i]; bunay boladi

// massivlarni chiqarishda cout<<a[i]; bolsa

// korsatkich shaklida cout<<\*(a+i); bunday boladi

// makroslar

// #define KOPAYTMA (a, b) (a) \* (b)

// int a,b,b,c; cin>>a>>b; c=KOPAYTMA(a,b); cout<<c<<endl; agar a=2 b=3 b-sa c=6 chiqadi;

// funksiyalar

// funksiya 2 xil bo'lib int tipi va void tipi

// void da hech nima qaytarilmaydi int da return qilib qaytaradi

// setfill('0') no'l bilan toldiradi agar son bolmasa

// bular uchun ham #include <iomanip> kutubhona kerak

// setw(2) 2 ta honagacha bu cout<<setfill('0')<<setw(2)<<endl;

// bular setprecision(verguldan keyin n ta) << fixed << a; ga ohshash.

// global ozgaruvchilar int main() dan tashqarida boladi va :: x tarzida chaqiriladi

// cout<<::x<<endl;

// korsatgichlar bilan ishlash

// n=5 -> nptr=&n -> nptr=0x22ff44

// n=15 \*nptr=n; -> \*nptr=15;

// korsatgich joylashgan adres &nptr=0x22fe30

// tasodifiy son uchun #include <stdlib.h> kutubhonasi qoshiladi.

// x=rand % n + 1; x-> n gacha bo'lgan tasodifiy son

// har doim har hil chiqarish uchun # include <ctime> kutubhona va srand(time(NULL)); qoshiladi

// s ni butundan keyin 3 ta qoldirib qoganini tashlab yuborish

// => printf("%.3f",s); buning uchun shu kutubhona kerak #include <stdio.h> #include <stdlib.h> kutubhona kerak

// 1000000 => 1E+6;

// switch(x){

// case 1: {break;}

// case 2: {break;}

// .... }

// a\*180/pi;

// ceil(x) => haqiqiy toifadagi x o'zgaruvchisi qiymatini unga eng yaqin katta burun songa aytiladi.

// floor(x) => haqiqiy toifadagi x o'zgaruvchisi qiymatini unga eng yaqin kichik burun songa aytiladi.

// #include <iomanip>

// setprecision(verguldan keyin n ta) << fixed << a; verguldan keyin x hona aniqlikda ekranga chiqarish

// continue; da oshi sonni qoldirib misolni davom etadi yani

// continue; dan keyin sikl tugab tashqaridagi sikl davom etadi.

// break; da esa siklni tugatadi

// Xarick\_\_07