

zadaci

zadatak	grupe	šifre	mapa
izvorni kôd	grupe.pas grupe.c grupe.cpp	sifre.pas sifre.c sifre.cpp	mapa.pas mapa.c mapa.cpp
ulazni podaci	standardni ulaz	biblioteka	standardni ulaz
izlazni podaci	standardni izlaz	biblioteka	standardni izlaz
memorijsko ograničenje (heap)	32 MB		
memorijsko ograničenje (stack)	1 MB		
vremensko ograničenje (pentium4 na 1.6 ghz)	4 sekunde	5 sekundi	6 sekundi
broj bodova	40	50	80
	170		

grupe

Na ploči je, slijeva na desno, u nekom proizvoljnom redoslijedu, zapisano **prvih N prirodnih brojeva**. Grupu definiramo kao skup brojeva koji zajedno čine neki uzastopni interval prirodnih brojeva. Npr. skupovi [2], [4 5] i [3 5 6 4] su grupe, ali [5 7 2] i [2 4 5] nisu.

Na početku, smatramo da svaki broj na ploči čini grupu u kojoj je on jedini element.

Jedina dozvoljena operacija je **spajanje dvije susjedne grupe**, ali **samo** ako će novonastali skup također imati svojstvo grupe.

Napišite program koji će odrediti **da li postoji** niz od $N-1$ operacija spajanja, nakon čijeg se izvršavanja **svi brojevi nalaze u istoj grupi** te, ako je odgovor potvrđan, **pronaći** jedan takav niz.

Primjer (jedno moguće rješenje za treći test primjer):

```
[6] [3] [2] [1] [4] [5]
[6] [3] [2 1] [4] [5]
[6] [3 2 1] [4] [5]
[6] [3 2 1] [4 5]
[6] [3 2 1 4 5]
[6 3 2 1 4 5]
```

ulazni podaci

U prvom retku se nalazi cijeli broj N , $1 \leq N \leq 500,000$.

U drugom retku se nalazi N prirodnih brojeva, međusobno odvojenih s po jednim razmakom, redom kojim su zapisani na ploči.

izlazni podaci

U prvi redak treba ispisati riječ 'DA' ili 'NE', ovisno o tome da li postoji traženi niz operacija.

Ako je ispisana riječ 'NE', onda ne treba više ništa ispisati.

Ako je ispisana riječ 'DA', onda u sljedećih $N - 1$ redaka treba ispisati po dva prirodna broja a i b tako da u $(i+1)$ -om retku a i b predstavljaju najmanji i najveći broj u grupi nastaloj i -tim spajanjem.

test primjeri

ulaz

```
2
2 1
```

izlaz

```
DA
1 2
```

ulaz

```
6
1 4 2 5 3 6
```

izlaz

```
NE
```

ulaz

```
6
6 3 2 1 4 5
```

izlaz

```
DA
1 2
1 3
4 5
1 5
1 6
```

šifre

Slavica je nedavno proslavila osamnaesti rođendan velikim tulumom i slike s istog je stavila na mrežu da ih vide i njeni prijatelji. Kako ipak ne želi da njena suparnica Mirjana vidi slike, ona ih je zaštitila tako da je svakom svom prijatelju dala šifru pomoću koje se može pristupiti slikama.

Mirjana bi jako željela pogledati slike i srećom zna da Slavičin program radi na sljedeći način:

Svaka šifra je niz **malih** slova engleske abecede, a u datoteci na disku se nalaze sve postojeće šifre. Kada netko pokuša pristupiti i upiše neki niz znakova kao šifru, program **uspoređi** upisani niz znakova sa **svakom** šifrom iz datoteke.

Da bi program usporedio dva niza znakova A i B potrebno je $K+1$ milisekundi, gdje je K **najveći broj** takav da je **prvih K znakova od A isto kao i prvih K znakova od B**. Na primjer, 'mirko' i 'slavko' se usporede za 1 milisekundu, 'ma' i 'mama' za 3 milisekunde, 'tata' i 'tata' za 5 milisekundi, a 'mirko' i 'mirjana' za 4 milisekunde.

Napišite program koji će pozivanjem funkcije **Probaj** pronaći barem jednu ispravnu šifru.

U datoteci se nalazi najviše 100 **međusobno različitih** šifri, svaka je duljine najviše 100 znakova, a vaš program treba odrediti jednu šifru s **najviše 1700** poziva funkcije **Probaj**.

biblioteka

Za rješavanje zadatka na raspolaganju je biblioteka funkcija **sifrelib** koja sadrži tri funkcije:

Init - ovu funkciju je potrebno pozvati **tačno jednom** i to **na početku** vašeg programa, bez argumenata. Ova funkcija ne vraća nikakvu vrijednost.

```
procedure Init;
```

```
void Init(void);
```

Probaj - ova funkcija se poziva s jednim argumentom, nekim nizom znakova. Funkcija vraća **ukupno vrijeme** (u milisekundama) potrebno da se prosljeđeni niz znakova uspoređi sa **svim** šiframa u datoteci. Niz znakova koji prosljeđujemo funkciji ne smije biti **prazan** string i mora se sastojati **samo** od malih slova engleske abecede. Nepoštivanje ovih ograničenja rezultirat će gubitkom bodova.

```
function Probaj(a : string) : longint;
```

```
int Probaj(const char *a);
```

Rjesenje - ovu funkciju je potrebno pozvati **na kraju** vašeg programa; kao argument prosljeđujete neku otkrivenu šifru. Ova funkcija ne vraća nikakvu vrijednost i **regularno završava izvođenje** vašeg programa.

```
procedure Rjesenje(a : string);
```

```
void Rjesenje(const char *a);
```

Upute za natjecatelje u Pascal-u

Na početku vašeg programa mora se nalaziti naredba 'uses sifrelib'.

Također, potrebno je preuzeti datoteke `sifrelib.o` i `sifrelib.ppu` s evaluacijskog sustava.

Upute za natjecatelje u C/C++

Potrebno je preuzeti datoteke `sifrelib.h` i `sifrelib.o` s evaluacijskog sustava.

U RHIDE-u otvorite projekt (Project->Open) i nazovite ga `sifre`. U njega naredbom Project->AddItem dodajte datoteke `sifre.c` odnosno `sifre.cpp` (vaš program) i `sifrelib.o`.

Na početku vašeg programa mora se nalaziti naredba '#include "sifrelib.h"'.

Važno: nemojte koristiti naredbu Compile->BuildAll (jer ona briše datoteku `sifrelib.o`).

Testiranje

Biblioteka koja vam je na raspolaganju čita sa standardnog ulaza i piše na standardni izlaz.

ulaz

2

slavica

mirko

izlaz

OK!

mapa

Jedan od većih problema u modernoj kartografiji je upisivanje imena zemljopisnih lokaliteta na mapu. Naime, kada se na mapu upiše naziv svakog grada, mjesta, sela i zaselka, onda se neki od tih naziva međusobno preklapaju pa je mapa nečitka.

U koordinatnom sustavu u ravnini su zadane točke koje je potrebno označiti. Prostor za naziv je **pravokutnik** čije su stranice **paralelne s koordinatnim osima** takav da je njegova **širina točno tri puta veća od visine**.

Za svaku točku mora postojati točno jedan prostor za naziv koji mora biti postavljen točno tako da se točka nalazi u njegovom **gornjem lijevom** uglu (tj. prostor za naziv neke točke se mora nalaziti neposredno dolje desno od nje).

Iz estetskih razloga svi prostori za naziv moraju biti **jednakih** dimenzija.

Napišite program koji će odrediti **maksimalnu moguću** veličinu prostora za naziv tako da se nakon što se svaka točka označi niti jedna dva prostora za naziv međusobno **ne preklapaju** (smiju se dodirivati bridom ili vrhom).

Dozvoljeno je da pravokutnici izlaze iz okvira 1,000,000 x 1,000,000 u kojem se nalaze točke.

ulazni podaci

U prvom retku se nalazi cijeli broj N , $2 \leq N \leq 100,000$, broj točaka na mapu.

U svakom od sljedećih N redaka se nalaze po dva cijela broja X i Y , $0 \leq X, Y \leq 1,000,000$, koordinate točaka. Sve točke će biti međusobno različite.

izlazni podaci

U prvi i jedini redak treba ispisati **visinu** traženog pravokutnika iz teksta zadatka, **realni broj zaokružen na 2 decimale**.

Zbog nepreciznosti aritmetike realnih brojeva, dopušteno odstupanje je **0.01**.

test primjeri

ulaz	ulaz	ulaz
3	5	10
0 0	1 1	26 77
5 4	6 5	12 37
3 7	18 3	14 18
izlaz	9 9	19 96
3.00	16 15	71 95
	izlaz	91 9
	4.00	98 43
		66 77
		2 75
		94 91
		izlaz
		7.67