

به نام خدا



درس مبانی برنامه‌سازی

تمرین ۶

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نیم سال اول ۹۹-۰۰

استاد:

رضا فکوری

مهلت ارسال:

۱۴ بهمن - ساعت ۲۳:۵۹:۵۹

مسئول تمرین‌ها:

امیرمهدی نامجو، پرهام صارمی

مسئول تمرین ۶:

فاطمه خاشعی

طراحان تمرین ۶:

مسیح بیگی، پرهام چاوشیان، آریا جلالی، علی حاتمی تاجیک، محمدرضا مفیضی

فهرست

سوالات

۲	سوال ۱. بازرس موریل
۲	سوال ۲. اعداد بزرگ
۵	سوال ۳. اکسیون در کلو تار
۷	سوال ۴. اطلاعات محرمانه
۹	سوال ۵. نایروبی
۱۴	سوال ۶. شناسایی پروفیسور معروف
۱۹	



سوالات

سوال ۱. بازرس موریل

پس از اینکه خانه پروفیسور در شهر پالومک (Palomeque) کشف شد، بازرس موریل به آنجا رفت تا از تمام اطلاعاتی که در آنجا یافت می‌شود استفاده کند. تخته سیاهی در میان خانه پیدا شد که چند کد به زبان C روی آن نوشته شده بود. بازرس موریل که نمی‌تواند از آنها سردر بیاورد کد ها را به تیم IT پلیس اسپانیا می‌سپارد اما چون با ماهیت اشاره‌گرها ناآشنا بودند از پس آنها بر نمی‌آیند. حال بازرس از شما خواسته است تا این قطعه کد ها را تحلیل کرده و نتایج آن را در سامانه کوئرا بارگذاری کنید.

قطعه کد اول

```
1  #include<stdio.h>
2
3  int main(){
4      int i = 5;
5      void *vptr;
6      vptr = &i;
7      printf("\nValue of iptr = %d ", *vptr);
8      return 0;
9  }
```

قطعه کد دوم

```
1  #include<stdio.h>
2
3  int main(){
4      char *cities[] = {"Helsinki", "Berlin", "Rio"};
5      char **i = &cities[0];
6      char **j = &cities[1];
7      char **k = &cities[2];
8      printf("%c%c%c\n", **i,**j,**k);
9      return 0;
10 }
11
```



قطعه کد سوم

```
1  #include<stdio.h>
2  #include<stdlib.h>
3
4  int main()
5  {
6      int *p;
7      container = (int *)malloc(20);
8      printf("%d\n", sizeof(container));
9      free(p);
10     return 0;
11 }
```

قطعه کد چهارم

```
1  #include<stdio.h>
2
3  int main()
4  {
5      char *a = "Respuesta incorrecta";
6      printf("%creías haber descubierto el plan?"
7            "%s la posición se perdió!" + 1
8            ,a
9            ,"Lorem Ipsum \0\b dolor sit amet");
10 }
11
```

قطعه کد پنجم

```
1  #include<stdio.h>
2  #include <string.h>
3
4  int main()
5  {
6      char *str = "Que";
7      int i;
8      for(i = 0; i < strlen(str); i++)
9          printf("%s", str++);
10     return 0;
11 }
12
```



قطعه کد ششم

```
1  #include<stdio.h>
2
3  int main()
4  {
5      int i = 22265484213,*pointer;
6      pointer= &i;
7      void *void_ptr;
8      void_ptr = &pointer;
9      printf("\nValue of iptr = %d ", **(int *****))
void_ptr);
10     return 0;
11 }
12
```

قطعه کد هفتم

```
1  #include<stdio.h>
2
3  int main()
4  {
5      char *string;
6      string = "Le Casa de Papel";
7      *(str+1) = 'a';
8      return 0;
9  }
10
```



سوال ۲. اعداد بزرگ

نایروبی موقع چاپ کردن پول‌ها متوجه میشه که دستگاه اول n_1 تا سکناس یک دلاری چاپ کرده و دستگاه دوم n_2 تا. به دلایل امنیتی متعدد نایروبی مجبور میشه این عددها رو تو یک مبنای دیگه نگه داره تا یک وقت لو نره. ولی حالا که خیالش راحت، متوجه میشه که فکر این جاشو نکرده بود و نمیتونه حساب کنه که کلا چقدر پول داره. در این عملیات به او کمک کنید. (شاید اگر کار را خوب انجام دهید مقداری از پول به شما برسد)

ورودی

ورودی در ۲ خط به شما داده میشود، در خط اول مبنای r و در خط بعد اعداد a_1 و a_2 به شما داده میشود. تضمین میشود اعداد داده شده در مبنای مورد نظر می‌باشند و نیازی به تبدیل مبنای نیست.

$$0 \leq a_1, a_2 \leq 10^{10000}$$

$$2 \leq r \leq 10$$

خروجی

خروجی برنامه‌ی شما باید شامل حاصل جمع ۲ عدد اول در مبنای داده شده باشد.

مثال

ورودی نمونه ۱

```
1 10
2 64755952212049917233 8084943563
3
```

خروجی نمونه ۱

```
1 64755952220134860796
2
```



ورودی نمونه ۲

```
1 2
2 1111110 10
3
```

خروجی نمونه ۲

```
1 10000000
2
```



سوال ۳. اکسیون در کلوتار

پروفسور قصد دارد برای سرقت ضرابخانه گروهی تشکیل دهد، او برای این کار یک بازی طراحی کرده تا باهوش‌ترین افراد را گزینش کند. ولی الان در به در در بنگاهی‌های اسپانیا، دنبال یک مقرر سری است و از شما می‌خواهد که برنامه‌ای بنویسید تا برنده بازی را در کمترین زمان و گرفتن کمترین مقدار حافظه مشخص کند.

بازی خیلی ساده است، کسی که کوچکترین عدد غیر تکراری را پیشنهاد دهد برنده بازی خواهد بود.

هرکس یک شماره شناسایی دارد که توسط پروفسور برایش ارسال می‌شود و کسی از کد شناسایی دیگری اطلاع ندارد. هر کس کد شناسایی‌اش را با عدد پیشنهادی‌اش به یک ایمیل ارسال کرده و جمع ایمیل‌ها ورودی برنامه خواهند بود.

البته توجه کنید که چون تمام شرکت‌کننده‌های این بازی از صنف زحمتکش دزدها، زورگیرها، آدمکش‌ها و ... هستند، ممکن است تقلب رخ دهد، یعنی یک فرد دو یا چند عدد پیشنهاد دهد در این صورت اگر کسی که چند عدد پیشنهاد داده برنده شد باید به عنوان متقلب گزارش شود اما اگر با تقلب هم نتوانست برنده شود لازم نیست اسمش ذکر شود و گزارش برنده کفایت می‌کند.

همچنین توجه داشته باشید که این بازی ممکن است برنده‌ای نداشته باشد.

ورودی

در هر خط یک شماره شناسایی و یک عدد (هر دو عدد طبیعی و کمتر از ۱,۰۰۰,۰۰۰,۰۰۰) با یک فاصله بینشان وارد می‌شوند. تعداد خط‌های ورودی نامشخص است و با آمدن عدد "۱ - تمام می‌شود".

خروجی

شماره شناسایی شخص برنده مثلاً در صورت برد (۸۱۳) :

813 won.

اگر کسی با تقلب برنده شد (مثلاً ۱۰۰۳) خروجی به این صورت خواهد بود:

1003 cheated.

در صورتی که کسی برنده نشد عبارت no one won چاپ شود.



مثال

ورودی نمونه ۱

```
1 10 2
2 11 13
3 12 2
4 13 10
5 -1
6
```

خروجی نمونه ۱

```
1 13 won.
2
```

ورودی نمونه ۲

```
1 10 20
2 11 3
3 12 4
4 13 3
5 11 2
6 -1
7
```

خروجی نمونه ۲

```
1 11 cheated.
2
```

ورودی نمونه ۳

```
1 10 3
2 11 3
3 -1
4
```

خروجی نمونه ۳

```
1 no one won.
2
```



سوال ۴. اطلاعات محرمانه

برلین طی تحقیقاتش به اطلاعات جدیدی در مورد دستگاه‌های ضراب‌خانه ملی پول اسپانیا دست یافته است اما موقع بررسی این اطلاعات متوجه می‌شود که همه اطلاعات به صورت رمز درآمده‌اند و بدون شکستن رمزشان قابل استفاده نیستند. چون برلین کارهای بسیار زیادی برای انجام دادن دارد، نمی‌تواند که وقتش را صرف شکستن کد اطلاعات بکند از این رو به سراغ ریو رفته و این وظیفه را به او می‌سپارد.

ریو با مشاهده اطلاعات متوجه می‌شود که با مجموعه‌ای از اعداد صحیح و کاراکترها مواجه است که برای بررسی لازم است ابتدا به فرم یک ماتریس سه‌بعدی در بیایند و سپس با اعمال تغییراتی به یک ماتریس دوبعدی تبدیل شوند که این ماتریس حاوی تمامی اطلاعات مورد نیاز برلین است. ریو برای انجام این تبدیلات دست به کار می‌شود اما متأسفانه متوجه می‌شود که کلنل تامایو و همکارانش نیز در این مدت بی‌کار نبوده‌اند و اختلالاتی در سیستم ریو ایجاد کرده‌اند. به دلیل این اختلالات ریو نمی‌تواند تابع‌هایی با بیش از یک ورودی بنویسد، همچنین به خاطر همین اختلالات ریو مجبور است که تمام اطلاعات ورودی را دقیقاً در یک ماتریس ذخیره کند و تمام اطلاعات خروجی را نیز در قالب یک ماتریس دیگر ذخیره و به برلین تحویل دهد. ریو که با دیدن این اختلالات شدیداً مستأصل شده است با شما تماس گرفته و دست به دامن شما می‌شود تا آبرویش را پیش برلین حفظ کنید. شما نیز تصمیم می‌گیرید که این لطف را در حق ریو انجام دهید و از او می‌خواهید که اطلاعات را برایتان ارسال کند.

وقتی اطلاعات به دستتان می‌رسد متوجه می‌شوید که در خط اول سه عدد n و m و l به ترتیب آمده است که اندازه سطرها اندازه n ، ستون‌ها m و اندازه بعد سوم $l+1$ است. سپس در ادامه $2nm$ خط آمده است. اگر شماره یکی از خطوط زوج باشد تنها یک کاراکتر T یا F آمده است که در نقشه ورودی باید به صورت boolean ذخیره بشود. اگر شماره خط فرد باشد بسته به اینکه خط بالایی‌اش T یا F باشد، در این خط l عدد صحیح یا کاراکتر آمده است که بعد سوم ماتریس ورودی را تشکیل می‌دهد. در واقع اگر در خط اول T بیاید یعنی در بعد سوم خانه $(1, 1)$ تعدادی عدد صحیح باید قرار بگیرد. دقت کنید که نحوه قرارگیری اطلاعات به گونه‌ای است که ابتدا هر سطر پر شده و سپس به سراغ سطر بعدی می‌رویم. در واقع $2m$ خطی که پس از خط اول می‌آیند همگی مربوط به صورت اول‌اند، $2m$ خطی که بعد از آن‌ها می‌آید، مربوط به سطر دوم است، و به همین ترتیب تا انتها پیش می‌رود.



هر خانه جدول نهایی به این صورت پر می‌شود که اگر بعد سوم متناظر با آن خانه در ماتریس ورودی شامل تعدادی عدد صحیح باشد، مقدار آن خانه در جدول نهایی برابر با بیشینه‌ی مجموع اعداد از بین تمام زیر دنباله‌های متوالی اعداد صحیح موجود در بعد سوم است. (دقت کنید که زیر دنباله تهی نیز یک زیر دنباله محسوب می‌شود!) اما اگر بعد سوم متناظر با آن خانه در ماتریس ورودی شامل تعدادی کاراکتر باشد، مقدار آن خانه در جدول نهایی برابر با کاراکتری است در مرتب‌سازی الفبایی پس از همه ظاهر می‌شود. اگر این کاراکتر یکی از حروف الفباست باید به صورت `case lower` باشد.

دقت کنید با توجه به اینکه جواب شما قرار است برای ریو ارسال شود و او باید کد شما را اجرا کند، لازم است تمام محدودیت‌هایی را که برای سیستم ریو به وجود آمده رعایت کنید تا او از شما ناامید نشود. همچنین حتما باید ماتریس‌های توضیح داده شده (ماتریس اطلاعات ورودی و اطلاعات نهایی) را تشکیل دهید تا در صورت لزوم ریو بتواند از آنها استفاده کند. در صورت عدم انجام اینکار اون غمگین خواهد شد.

دقت کنید که اگر ریو را غمگین یا ناامید کنید او با هک کردن کوئرا نمره نهایی شما را به صفر تغییر خواهد داد!



ورودی

ورودی درواقع همان اطلاعاتی است که ریو برای شما فرستاده است و در همان قالب است.

$$1 \leq n, m, l \leq 100$$

خروجی

کافی است که ماتریس دو بعدی ساخته شده را در خروجی چاپ کنید. لازم است که اعداد و کاراکترهای هم‌سطر به وسیله space از هم جدا بشوند.

مثال

ورودی نمونه ۱

```
1 3 2 1
2 T
3 3
4 T
5 -1
6 F
7 a
8 T
9 5
10 F
11 S
12 F
13 *
14
```

خروجی نمونه ۱

```
1 3 0
2 a 5
3 s *
4
```

ورودی نمونه ۲

```
1 1 1 3
2 T
3 1 2 3
4
```

خروجی نمونه ۲



1 6

2

ورودی نمونه ۳

1 3 1 3

2 T

3 1 0 1

4 F

5 a B c

6 F

7 P A R

8

خروجی نمونه ۳

1 2

2 c

3 r

4

ورودی نمونه ۴

1 1 3 3

2 T

3 -1 -2 -3

4 T

5 1 2 3

6 F

7 F 0 P

8

خروجی نمونه ۴

1 0 6 p

2

ورودی نمونه ۵

1 2 2 2

2 T

3 1 -1

4 T

5 -1 -1

6 F

7 T A

8 F



9 C E
10

خروجی نمونه ۵

1 1 0
2 t e
3



سوال ۵. نایروبی

نایروبی در شماره‌گذاری اسکناس‌ها به مشکل برخوردیده است و به ذخیره‌سازی اعداد و ارقام مختلف نیاز دارد. آنها که تمام تلفن‌های همراه را خاموش کرده‌اند و نمی‌توانند از آنها برای فرایند ذخیره‌سازی استفاده کنند، برای همین مجبور به استفاده از رایانه داخل یکی از ماشین‌های چاپ قدیمی شده‌اند. ولی آنها دو مشکل دارند، اول اینکه ریو مشغول رسیدگی به دختر وزیر است و دوم، این رایانه حافظه محدودی دارد. آنها مشکلشان را با پروفیسور درمیان گذاشته‌اند.

حال شما از طرف پروفیسور ماموریت دارید تا برنامه بنویسید که نایروبی بتواند با آن به ذخیره اطلاعات خود بپردازد. این برنامه باید به شکل یک آرایه عمل کند و اطلاعات را درون خود نگه‌دارد. یکی از تفاوت‌های این برنامه با آرایه‌های معمولی در آن است که ما تعداد داده‌ای که قرار است در حافظه ذخیره کند را از قبل نمی‌دانیم. تفاوت دیگر آن است که هر تمام انواع داده‌ها (bool char float int) باید در این آرایه ذخیره شوند. نایروبی در هر لحظه باید بتواند با استفاده از ایندکس داده‌ای که وارد کرده به آن دسترسی داشته باشد، بتواند تمام اطلاعات را به یکباره ببیند و بتواند تمام داده‌ها را پاک کند.

این رایانه به خاطر قدیمی بودن از فرمت آرایه‌ها پشتیبانی نمی‌کند و شما قادر به استفاده از آرایه‌ها در برنامه‌تان نخواهید بود. همچنین محدودیت حافظه این رایانه را هم در نظر داشته باشید و از حافظه‌ای که در اختیارتان است به خوبی استفاده کنید.

نحوه وارد کردن اطلاعات به برنامه به این صورت است که نایروبی ابتدا نوع داده خود را وارد می‌کند. و سپس داده خود را به برنامه خواهد داد (تضمین می‌شود که نوع داده و ورودی آن داده همگی مقادیر قابل قبول باشند). برای چاپ تمام داده‌ها، دسترسی به خانه‌های حافظه و پاک کردن تمام داده‌ها نیز دستورات دیگری وجود دارد.



دستورات دریافت اطلاعات

ابتدا نوع داده مشخص می‌شود و در خط بعد از آن داده دریافت می‌شود. پیغام اضافه شدن داده به آرایه به این صورت زیر است:

ورودی

```
1 <type>[int/float/char/bool]
2 <data>
3
```

خروجی

```
1 <type>[Integer/Float/Character/Boolean] <data> added to index <index>
2
```

ورودی از نوع bool به صورت صفر یا یک دریافت می‌شود اما باید به صورت True و False نمایش داده‌شود.

دسترسی به یک عنصر خاص

ابتدا عبارت access وارد می‌شود و در خط بعد یک عدد صحیح به عنوان index دسترسی داده خواهد شد (معتبر بودن این مقدار تضمین نمی‌شود).

ورودی

```
1 access
2 <index>
3
```

خروجی

اگر عدد وارد شده معتبر باشد عنصر مورد نظر با فرمت زیر باید در کنسول چاپ شود.

```
1 <type>[Integer/Float/Character/Boolean] <data> is in index <index>
2
```

اگر عدد وارد شده معتبر نبود پیغام زیر باید چاپ شود.

```
1 Out of Boundaries
2
```




چاپ تمام اطلاعات

با این دستور تمام اطلاعات ثبت شده چاپ خواهند شد.

ورودی

```
1 print-all
```

```
2
```

خروجی

برای تمامی ها index (از صفر تا آخرین داده ثبت شده) ، باید یک خط شامل داده های زیر چاپ بشود.

```
1 <index>: <type>[Integer/Float/Character/Boolean] - <data>
```

```
2
```

اگر داده ای در آرایه وجود نداشت، پیغام زیر نمایش می یابد.

```
1 Nothing Exists...
```

```
2
```

پاک کردن اطلاعات

با این دستور تمام پاک می شوند و شماره گذاری آرایه از صفر آغاز خواهد شد.

ورودی

```
1 erase-all
```

```
2
```

خروجی

```
1 All Data Has Been Cleared...
```

```
2
```

خروج از برنامه

کاربر با دستور زیر از برنامه خارج می شود. در کنسول چیزی چاپ نخواهد شد.

```
1 out
```

```
2
```



مثال ورودی نمونه

```
1  int
2  5
3  float
4  2.365
5  char
6  s
7  bool
8  1
9  access
10 20
11 access
12 2
13 print all
14 erase all
15 print all
16 int
17 5
18 out
19
```

خروجی نمونه

```
1 Integer 5 added to index 0
2 Float 2.365000 added to index 1
3 Character s added to index 2
4 Boolean True added to index 3
5 Out of Boundaries
6 Character s is in index 2
7 0: Integer - 5
8 1: Float - 2.365000
9 2: Character - s
10 3: Boolean - True
11 All Data Has Been Cleared...
12 Nothing Exists...
13 Integer 5 added to index 0
14
```



نکات

- از حافظه‌ای که در اختیارتان قرار گرفته است به خوبی استفاده کنید.
- شما در این تمرین مجاز به استفاده از براکت [] نیستید.
- اعداد اعشاری را تا شش رقم اعشار چاپ کنید.



سوال ۶. شناسایی پروفسور معروف

این تمرین نمره ای ندارد و برای علاقه‌مندان طرح شده است.
بازرس از قبرستان ماشین‌ها نمونه خونی (که از شکستگی دماغ پروفسور در ون ریخته بود) پیدا کرده است و آن را برای آزمایش‌های بیشتر به آزمایشگاه نزد آلبرتو (Alberto Vicuña) می‌فرستد. آلبرتو نمونه dna را خارج می‌کند اما برای آنالیز آن نیاز به یک نرم‌افزار خاص دارد که شما باید آنرا طراحی و در اختیار پلیس اسپانیا قرار دهید.

از خونی که پیدا شده است dna استخراج شده که باید با یک الگوی آنالیزی، احتمال تطابق dna با مجرم مشکوک را پیدا کنیم. برای این کار قطعه ای از dna به طول ۱۰ کاراکتر و الگوی آنالیزی که یکی از الگوهای AAA یا A۲A یا A۴A است و ضریب کاتالیزور برای انجام آزمایش روی آن که به صورت صفر یا یک (صفر یعنی کاتالیزور صعودی و یک نزولی) است داده می‌شود.

نکته اینجاست که نمی‌توان به صورت مستقیم روی قطعه dna ی که در صحنه جرم پیدا شده آنالیز انجام داد. برای همین باید حتما فرایندی روی آن انجام شود تا خروجی آماده برای آنالیز (نمونه) بدست آید. این فرایند همان تابع prepare است که در آن کاتالیزور صعودی یا نزولی استفاده می‌شود.

پس ابتدا باید با استفاده از تابع prepare نمونه آماده برای آنالیز را دریافت کرد. سپس نمونه را به همراه تابع آنالیز که توسط شما پیاده سازی می‌شود به تابع probability داده تا احتمال تطابق، که همان خروجی نهایی است را به شما برگرداند.



توابعی که باید شما پیاده سازی کنید:

- توابع کاتالیزور: (اسم این تابع دلخواه است)

```
1 int catalisor(const void *base, const void *other);
```

این تابع باید به صورت مجزا برای هر ضریب (صفر و یک) پیاده‌سازی شود. این توابع باید مقایسه دو داده کاراکتری را انجام دهند به این صورت که اگر کاتالیزور صعودی بود و ورودی اول بزرگتر بود ۱، اگر ورودی دوم بزرگتر بود ۱- و در غیر این صورت ۰ برگرداند. کاتالیزور نزولی هم به صورت عکس عمل می‌کند.

- توابع آنالیز:

این تابع باید به صورت مجزا برای هر ضریب (صفر و یک) پیاده‌سازی شود این تابع باید به صورت مجزا برای هر الگو پیاده‌سازی شود. به این صورت که با توجه الگو آنالیز را روی عناصر نمونه انجام دهد و نتیجه‌ها را در آرایه ای بریزد. خروجی این تابع اشاره‌گری به آرایه نتایج است.



توابعی که باید از آن استفاده کنید:

• تابع `prepare` :

```
1 void **prepare(char *, int (*)(const void *, const void *));
```

```
2
```

ورودی :

○ قطعه dna

○ اشاره‌گری به تابع کاتالیزور مناسب

این تابع قطعه dna و اشاره‌گری به تابع کاتالیزور ورودی می‌گیرد. در خروجی اشاره‌گری به نمونه آماده شده برمی‌گرداند. نکته مهم این است که این نمونه، آرایه ای هم اندازه با قطعه dna ورودی است. (یعنی نمونه آرایه ای به طول ۱۰ می‌باشد)

• تابع `get_analyzer` :

```
1 void *get_analyzer();
```

```
2
```

خروجی :

○ اشاره‌گری به تابع آنالیزور

این تابع اشاره‌گری به تابع آنالیزور که در حین فرایند آنالیز نمونه استفاده می‌شود برمی‌گرداند.

• تابع آنالیزور : (اسم این تابع دلخواه است)

```
1 int analyzer(void *one, void *other);
```

```
2
```

این تابع با گرفتن دو ورودی (که در واقع دو عنصر از آرایه نمونه می‌باشد) نتیجه آنالیز را به صورت یک عدد برمی‌گرداند.



• تابع probability :

```
1 float probability(void **, void (*)(void **), int);  
2
```

ورودی :

- اشاره‌گری به نمونه
 - اشاره‌گری به تابع آنالیز (توجه کنید که برای هر الگو تابع آنالیز مجزایی را باید به این تابع ورودی دهید.)
 - طول خروجی تابع آنالیز
- این تابع عملیات اصلی آنالیز را بر روی نمونه انجام می‌دهد و در خروجی احتمال تطابق dna را با مجرم می‌دهد.

الگوها:

• الگوی AAA :

در این الگو باید با استفاده از تابع آنالیزور هر عنصر از آرایه نمونه را با عنصر بعدی به تابع آنالیز دهید. (خروجی آنالیز این الگو آرایه ای به طول ۹ است.)

• الگوی A۲A :

در این الگو باید با استفاده از تابع آنالیزور یک عنصر از اول آرایه نمونه را با عنصری مقابل آن از آخر آرایه به تابع آنالیز دهید. (مثلاً ۰ با ۹ و ۱ با ۸) (خروجی آنالیز این الگو آرایه ای به طول ۵ است.)

• الگوی A۴A :

در این الگو باید هر دو آنالیز الگوهای AAA و A۲A را انجام دهید. (یعنی نتایج الگوی A۲A را بعد از نتایج الگوی AAA در آرایه بریزد.) (خروجی آنالیز این الگو آرایه ای به طول ۱۴ است.)



ورودی:

- قطعه dna به صورت آرایه ای از کارکتر به طول ۱۰ در یک خط
- الگوی آنالیز (یکی از الگوهای AAA و A۲A و A۴A) در یک خط
- ضریب کاتالیزور به صورت یک عدد صفر یا یک (صفر یعنی صعودی و یک نزولی) در یک خط

خروجی:

در نهایت باید احتمال تطابق را با دقت دو اعشار در خروجی چاپ کنید.

نمونه ورودی:

```
1 ACGGGTGTTT
2 A4A
3 1
4
```

در این جا قطعه dna وارد شده ACGGGTGTTT را به همراه کاتالیزور صعودی به تابع prepare می‌دهیم. بعد خروجی این تابع را به همراه تابع آنالیز با الگوی A۴A و عدد ۱۴ به عنوان طول خروجی آنالیز به تابع probability می‌دهیم.

نمونه خروجی:

```
1 0.36
2
```

تابع probability در خروجی مقدار ۰.۳۶ می‌دهد.