

به نام خدا



درس برنامه نویسی پیشرفته

تمرین سری اول

دانشکده مهندسی کامپیوتر

دانشگاه علم و صنعت

استاد مرضیه ملکی مجد

نیم سال دوم ۱۴۰۱-۱۴۰۲

مهلت ارسال:

۱۴۰۰/۱۲/۰۵

مبحث:

آشنایی با زبان C#

مسئول تمارین:

آریا شهسوار

فهرست:

- آداب نامه تمرینات-----۳
- نکات تمرینات سری اول-----۴
- سوال ۱:رمزنگاری-----۵
- سوال ۲:اعداد فیبوناچی اول-----۷
- سوال ۳:چاپ دو مثلث روبروی هم-----۸
- سوال ۴:گرد کننده-----۹
- سوال ۵:Maximum and-----۱۱

آداب نامه تمرینات:

- پاسخ تمامی سوالات فقط با زبان C# قابل قبول است.
- علیرغم اعتماد کامل تیم تی ای به شما دانشجویان، تمامی کد های شما به صورت خودکار و توسط برنامه بررسی خواهد شد. همچنین در طول ترم از تمامی کد های شما ارائه گرفته خواهد شد و نحوه کار تمامی بخش های کد از شما گرفته پرسیده خواهد شد. لذا از کپی کردن کد دوستان تان خودداری کنید.
- از آنجایی که مشورت و هم فکری با دوستان تان کار بسیار پسندیده ای میباشد برخلاف تقلب:) در صورت همفکری با دانشجو یا دانشجویانی نام آنها را در ابتدای کد خود به صورت کامنت شده بنویسید.
- برای ارسال تمرین در طول ترم مجموعاً ۱۰ روز تاخیر مجاز دارید که حداکثر مقدار قابل استفاده برای هر تمرین ۳ روز از آن ۱۰ روز میباشد.
- پس از پایان این ۱۰ روز، دیگر تمرین ارسالی در صورت تاخیر قابل قبول نخواهد بود و نمره تمرین در این صورت صفر در نظر گرفته خواهد شد.
- برای استفاده از تاخیر در هر سری تمرین، درخواست خود را پیش از اتمام مهلت ارسال تمرین به تی ای مربوطه اعلام نمایید.
- برای هر سری تمرین، تلاش شده است که سوالات ساده تر در ابتدا و سوالات پیچیده تر در انتها قرار بگیرد.
- در صورت وجود هرگونه ابهام یا سوال در مورد سوالات تمرین، سعی کنید سوال خود را تا جای ممکن در گروه پرسید. شاید سوال شما، سوال دوستان دیگر تان هم باشد و دوستان تان هم بتوانند از پاسخ شما بهره ببرند.

نکات تمرین سری اول:

- سوالات را در کوئرا و در بخش تمرین سری اول آپلود کنید.
- در سوال دوم بررسی اول بودن اعداد باید به روش بازگشتی صورت گیرد.
- از آنجایی که هر سوال توسط تی ای خاصی طراحی شده است ، در صورت نیاز به پرسش راجع به سوالات طراح هر سوال بهتر از بقیه میتواند شما را راهنمایی کند. بنابراین در زیر اسامی طراحان هر سوال ضمیمه گردیده است.

- سوال ۱ : آقای اوشنی
- سوال ۲ : خانم کاووسی نژاد
- سوال ۳ : خانم کاووسی نژاد
- سوال ۴ : آقای زارعیان
- سوال ۵ : آقای زارعیان

تمرین ۱: رمزنگاری

شخصی در سازمانی سری کار میکند. وظیفه این شخص این است که پیام هایی را که باید ارسال کند رمزنگاری کند. این پیام ها همیشه به صورت یک عبارت دو کلمه ای میباشد. برای حفظ امنیت هرچه بیشتر این کلمات با استفاده از تابع زیر تبدیل به دو عدد میشوند. عدد *help* هم در ورودی داده میشود.

$$Changer(word) = \sum_{i=0}^{length\ of\ word} word[i] * (help)^i$$

سپس در خروجی مجموع ب.م.م و ک.م.م دو عدد به دست آمده را به عنوان پیام رمز شده نشان میدهیم.

در ورودی در خط اول عدد *help* و در خط دوم عبارت دو کلمه ای ما می آید.

نکته: کلمات با یک فاصله از یکدیگر جدا میشوند.

نکته: منظور از *word[i]* کد اسکی معادل حرف *i* ام آن میباشد.

مثال 1 :

Input:

4

salam refigh

Output:

5146503275

توضیح مثال 1:

Changer(refigh)==141734

Changer(Salam)==36311

GCD(141734, 36311)==1

LCM(141734, 36311)==5146503274

Output==GCD+LCM==5146503275

تمرین ۲: اعداد فیبوناچی اول

برنامه ای بنویسید که با گرفتن عدد n به عنوان ورودی، اعداد کوچکتر مساوی n که هم فیبوناچی و هم اول باشند را چاپ کند.

نکته: بررسی کردن اول بودن یا نبودن اعداد باید به صورت بازگشتی انجام شود.

مثال 1:

Input:

300

Output:

2 3 5 13 89 233

توضیح مثال 1:

سری فیبوناچی کوچکتر از 300 به صورت زیر است:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233

اعداد اول این سری:

2, 3, 5, 13, 89, 233

مثال 2:

Input:

40

Output:

2 3 5 13

تمرین ۳: چاپ دو مثلث روبه روی هم

برنامه ای بنویسید که با استفاده از توابع بازگشتی دو مثلث قائم الزاویه متساوی الساقین را روبه روی هم چاپ کند. ارتفاع مثلث در ورودی داده می شود.

مثال 1:

Input:

5

Output:

```
*
*
* *
* * *
* * * *
* * * * *
```

مثال 2:

Input:

2

Output:

```
*
* *
* * *
```


تمرین ۴: گرد کننده

به دلیل مشکلات اقتصادی یک فروشگاه، رییس فروشگاه قصد دارد تا قیمت کالا های خود را افزایش دهد. قیمت حال حاضر یک کالا را با " n " نمایش می‌دهیم. رییس فروشگاه میخواهد قیمت کالا های خود را k برابر کند ($1 \leq k \leq m$) به طوری که قیمت جدید کالا به گرد ترین شکل ممکن باشد. در ادامه تعریف گرد بودن توضیح داده شده است .

گرد بودن : میگوییم عدد a از عدد b گرد تر است هر گاه a تعداد صفر های بیشتری در سمت راست خود داشته باشد.

مثال: عدد 5120000 از عدد 100020 گرد تر است.

توجه: اگر دو عدد تعداد صفر های برابری در سمت راست خود داشته باشند عددی گرد تر است که بزرگتر باشد.

ورودی: در یک خط به ترتیب n و m را از ورودی میگیریم. n برابر با قیمت اولیه کالا است و m به معنای آن است که قیمت جدید کالا حداکثر برابر $m * n$ است.

$$1 \leq m, n \leq 10^9$$

خروجی: قیمت جدید کالا که مضربی از n است و گرد ترین عدد در بازه ی n تا $n * m$ میباشد.

توجه: اگر امکان آن که قیمت کالا گردتر بشود وجود نداشت $n * m$ را در خروجی نمایش دهید.

نمونه ورودی و خروجی در صفحه بعد آمده است.

مثال 1:

Input:

6 11

Output:

60

مثال 2:

Input:

10050 12345

Output:

120600000

مثال 3:

Input:

1 7

Output:

7

تمرین ۵: maximum And

به شما دنباله ی a از اعداد صحیح مثبت به طول n داده میشود.

باید در حداکثر k مرحله هر بار یک عدد از دنباله a را انتخاب کنید و آن را با 2^j ، Or کنید.

$$(0 \leq j \leq 30)$$

این کار را باید به گونه ای انجام دهید که در پایان حاصل And تمامی عناصر این دنباله بیشترین مقدار شود.

نکته: j دارای مقداری دلخواه است.

توجه: منظور از عملیات And و عملیات Or در بالا عملیات های bitwise است.

ورودی: در خط اول ورودی n و k به ترتیب داده میشود. (k تعداد مراحل است)

$$0 \leq k \leq 10^9$$

$$1 \leq n \leq 2 * 10^5$$

در خط بعدی n عدد با یک فاصله از هم می آیند که همان عناصر دنباله ی a هستند.

$$0 \leq a_i < 2^{31}$$

خروجی: طبق توضیحات حداکثر k عملیات را به گونه ای انجام دهید که حاصل And تمامی عناصر بیشترین مقدار شود و آن را در خروجی نمایش دهید.

توجه: تمامی ورودی ها و خروجی به صورت دسیمال است.

نمونه ورودی و خروجی در صفحه بعد آمده است.

مثال 1 :

Input:

3 2
2 1 1

Output:

2

توضیح مثال 1 :

$a[0] = 2$, $a[1] = 1$, $a[2] = 1$

مرحله ی اول :

$a[1] = a[1] \text{ or } 2^1 = 3$

مرحله ی دوم :

$a[2] = a[2] \text{ or } 2^1 = 3$

جواب نهایی :

2 And 3 And 3 = 2

مثال 2 :

Input:

1 30

0

Output:

2147483646

توضیح مثال 2 :

$a[0] = 0$

مرحله ی اول :

$a[0] = a[0] \text{ or } 2^{30} = 2^{30}$

مرحله ی دوم :

$a[0] = a[0] \text{ or } 2^{29} = 2^{30} + 2^{29}$

مرحله ی سوم :

$a[0] = a[0] \text{ or } 2^{28} = 2^{30} + 2^{29} + 2^{28}$

.

.

مرحله سی ام :

$a[0] = a[0] \text{ or } 2^1 = 2^{30} + 2^{29} + 2^{28} + \dots + 2^2 + 2^1$

و چون فقط یک عدد داریم خروجی برابر با $a[0]$ است.

موفق باشید :) .