

فاکتوریل

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

چرا تابع؟

توابع مهم‌ترین جزء برنامه‌نویسی رویه‌ای (Procedural Programming) هستند. تصور کنید در برنامه‌ای بخواهیم مرتباً اول بودن یا نبودن اعداد متفاوتی را بررسی کنیم. نوشتن کدی برای بررسی این قضیه درون کد اصلی برنامه نه تنها موجب تکرار بی‌رویه کد می‌شود بلکه از خوانایی کد نیز می‌کاهد. در برنامه‌نویسی عملیات‌هایی که مرتباً صورت می‌پذیرند را در قالب توابعی برای استفاده در جای‌جای برنامه تعریف می‌کنیم.

تابع main؟!

تا به امروز تمامی دستورات برنامه‌تان را درون تابع main می‌نوشتید. بله! main یک تابع است. تنها تفاوت تابع `main()` با سایر توابع آن است که این تابع نقطه شروع برنامه شما بوده و سیستم‌عامل به هنگام اجرای برنامه شما این تابع را فراخوانی (Call) می‌کند.

شیوه تعریف تابع!

تعریف هر تابع از دو بخش اصلی تشکیل می‌شود؛ بخش اولیه که Prototype یا ضابطه تابع نام دارد به صورت زیر تعریف می‌شود:

```
return_type functionName (parameters)
```

نخستین بخش از ضابطه تابع `return_type` آن است که نوع خروجی تابع را تعیین می‌کند. توابع می‌توانند داده‌ای همچون عددی صحیح یا اعشاری را به عنوان نتیجه پردازش خود به محل فراخوانی بازگردانند. نوع این داده را می‌بایست در این بخش تعیین کنید. اگر تابع شما هیچ خروجی ندارد از `void` استفاده کنید.

پس از آن نوبت به نام‌گذاری تابع می‌رسد. نام‌گذاری توابع نیز قواعدی مانند نام‌گذاری متغیرها دارد. متداول‌ترین شیوه نام‌گذاری توابع در زبان C سبکی به نام Camel Case است. در این سبک نام با حرف کوچک آغاز شده و هر جا که نیاز به جداسازی کلمات بود از حرف بزرگ استفاده می‌شود. به طور مثال تابعی که عددی دریافت کرده و اول بودن آن را بررسی می‌کند به صورت `isPrime` نام‌گذاری می‌کنیم.

پس از نام‌گذاری، نوبت به تعیین متغیرهای ورودی تابع می‌شود که درون پرانتزها تعریف می‌شوند. این متغیرها را به هنگام فراخوانی تابع تعیین می‌کنید و می‌توانید درون کد تابع از آن‌ها استفاده کنید. به طور مثال اگر تابعی به ترتیب یک عدد صحیح و دو عدد اعشاری را با نام‌های `integer` و `div` و `sum` دریافت کند پارامترهای آن را به صورت زیر تعریف می‌کنیم:

```
(int integer, float div, float sum)
```

در انتها نوبت به Implementation یا پیاده‌سازی تابع می‌رسد که همان دستورات مربوط به آن است. پیاده‌سازی تابع باید مابین `{ }` قرار گیرد.

توابع را کجا تعریف کنم؟

دو شیوه برای تعریف توابع وجود دارد:

- شما می‌توانید Prototype تمامی توابع (به جز `main`) را در ابتدای کد خود و درست پس از `#include` ها آورده و آن‌ها را با `;` تمام کنید و پس از آن توابع را به هر ترتیبی که خواستید تعریف کنید.

- در همان ابتدای کد خود ضابطه و پیاده‌سازی تابع را در کنار هم تعریف کنید.

برای روشن‌شدن این مسئله برنامه‌ای برای بررسی زوج یا فرد بودن عددی که از کاربر دریافت می‌شود را به هر دو صورت بررسی می‌کنیم. در شیوه اول می‌نویسیم:

```
#include <stdio.h>
```

```
int isEven(int number);
```

```
int main() {
    int x;
    scanf("%d", &x);
    printf("The number is %s.", isEven(x) ? "even" : "odd");
    return 0;
}

int isEven(int number) {
    return (number % 2 == 0);
}
```

و اما در شیوه دوم (که به علت خوانایی بالاتر مورد توصیه ماست):

```
#include <stdio.h>

int isEven(int number) {
    return (number % 2 == 0);
}

int main() {
    int x;
    scanf("%d", &x);
    printf("The number is %s.", isEven(x) ? "even" : "odd");
    return 0;
}
```

شیوه نوشتن توابع همچون موارد بسیار دیگری در برنامه‌نویسی سلیقه‌ای است. به هر سبکی که راحت‌ترید کد بنزید!

تمرین

برنامه‌ای بنویسید که عدد n را از ورودی گرفته و فاکتوریل آن را محاسبه کرده و نمایش دهد. (الزامی است در این برنامه از تابعی برای عمل فاکتوریل استفاده کنید.)

ورودی

در یک خط عدد n به شما داده می‌شود.

$$1 \leq n \leq 10$$

خروجی

فاکتوریل را در خروجی چاپ کنید.

مثال

ورودی نمونه

5

خروجی نمونه

120

تابع نمایی

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۱۲۸ مگابایت

برنامه‌ای بنویسید که مقدار e^x را با کمک رابطه‌ی زیر محاسبه کند (تا سه رقم اعشار). این برنامه مقدار x و تعداد جملات سری، n ، را از ورودی دریافت می‌کند.

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

ورودی

در خط اول عدد x و در خط بعدی عدد n به شما داده می‌شود.

$$0 \leq n, x \leq 1\,000$$

خروجی

در تنها خط خروجی مقدار e^x را با دقت سه رقم اعشار چاپ کنید.

مثال

ورودی نمونه ۱

5
10

خروجی نمونه ۱

143.689

ورودی نمونه ۲

3

15

خروجی نمونه ۲

20.086

کشف معادله (تابع)

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۵۰ مگابایت

برنامه ای بنویسید که با گرفتن n و مختصات n نقطه (به ترتیب x و y) اعلام کند همی این نقطه‌ها متعلق به کدام یک از نمودارها می‌توانند باشند و شماره‌ی نمودار را چاپ کند و اگر متعلق به هیچ کدام نبود، عبارت No ones را چاپ کند. برای این که یک دسته نقطه متعلق به یک نمودار باشد، کافیست برای هر یک از n نقطه، اختلاف y نقطه با y نمودار، در آن نقطه کمتر یا مساوی ۰.۲ باشد. برای هرکدام از معادلات زیر، یک تابع جداگانه تعریف کنید.

- $y = x - [x]$
- $y = x^2 + x$
- $y = |-x^3 + 1| + x^3$

ورودی

در خط اول ورودی عدد n آمده است.

در n خط بعد در هر خط ۲ عدد گویا آمده که مختصات نقطه‌ها را مشخص می‌کنند. (قدر مطلق تمامی طول و عرض نقطه‌ها از ۱۰ بیشتر نیست).

$$1 \leq n \leq 10$$

خروجی

در خروجی اطلاعات خواسته شده را چاپ کنید.

مثال

ورودی نمونه ۱

3
0.5 0.6
0.4 0.5
1.1 2.4

خروجی نمونه ۱

2

ورودی نمونه ۲

2
0.0 1.1.3
1.0 0.7

خروجی نمونه ۲

No ones

ضيق وقت

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

من وقت ندارم براتون داستان بسازم. برنامه شما باید جمله n ام دنباله فیبوناتچی رو حساب کند، اما باید از تابع استفاده کنید و تابع شما **تنها** می تواند یک *سمیکالن* (;) داشته باشد.

ورودی

در تنها خط ورودی عدد n آمده است.

خروجی

در خروجی، جمله n ام دنباله فیبوناتچی رو چاپ کنید.

مثال

ورودی نمونه ۱

5

خروجی نمونه ۱

5

عدد به پیر به پیغمبر اول!!

قنبر (که مطمئنم الان همتون عاشقشین) همینطور به محبوبیتش اضافه شده و هر روز با ایده های جدید داره برای شما کار میتراشه: / و اما آخرین ایده ی اون این بوده: عددی که هم اول باشد و هم متقارن رو عدد "به پیر به پیغمبر اول" میگوین! . اما دوباره سوالات مردم سراریز شدن و قنبر که از جواب دادن به اونها عاجزه به سراغ شما اومده تا سوالاتشو بپرسه و اعدادو براش حساب کنید.

ورودی

ورودی شامل یک خط است که در آن دو عدد n و b به ترتیب با یک فاصله از یکدیگر آمده اند.

$$1 \leq n \leq 100$$

$$2 \leq b \leq 10$$

خروجی

خروجی برنامه شامل یک خط است که در آن n مین عدد به پیر به پیغمبر اول را به مبنای b برده و چاپ میکنید.

توجه کنید در این سوال باید از تابع استفاده کنید. یک تابع برای چک کردن برای اول بودن عدد، یک تابع برای معکوس کردن عدد و یک تابع برای تبدیل مبنا به b

اگر تابع های گفته شده پیاده نشده باشند نمره شما پس از تصحیح دستی کسر خواهد شد.

مثال

ورودی نمونه ۱

خروجی نمونه ۱

401