

اصول کامپیوتر ۱

مبانی کامپیوتر و برنامه‌سازی

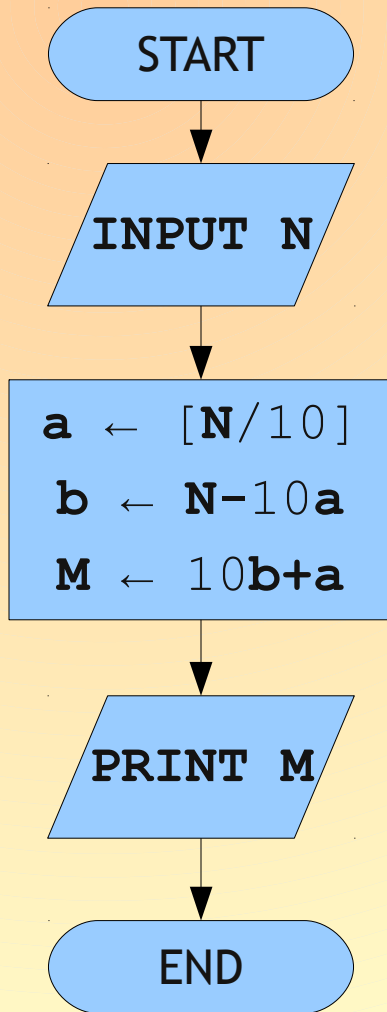
«جلسه‌ی هفتم»

دانشکده‌ی علوم ریاضی - دانشگاه شهید بهشتی

نیم‌سال اول ۹۰-۱۳۸۹

مدرس: سید علی کتان‌فروش

یادآوری



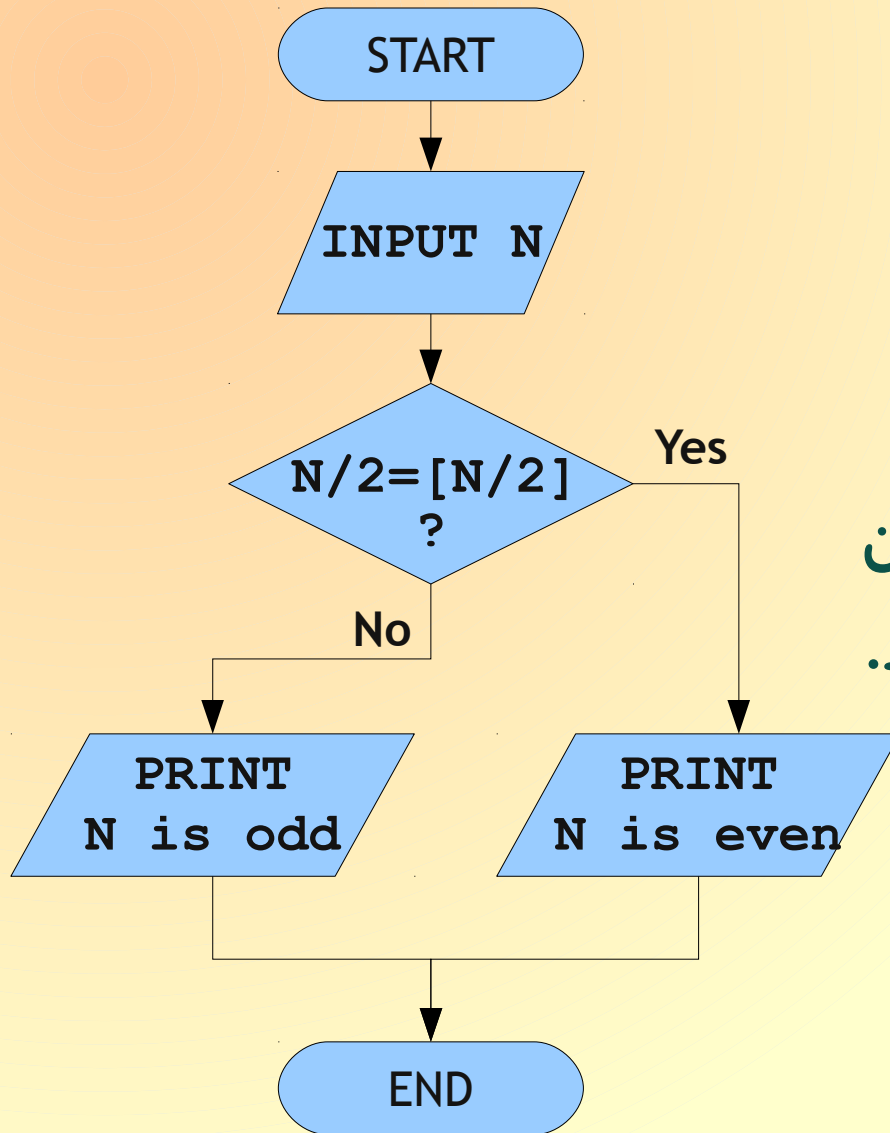
- ◆ در جلسه‌ی قبل با ساده‌ترین نوع از الگوریتم‌ها که آنرا الگوی «ورودی-محاسبه-نتیجه» می‌نامیدیم آشنا شدیم.
- ◆ در الگوریتم‌های «ورودی-محاسبه-نتیجه»، فلوچارت روالی خطی دارد و اصطلاحاً می‌گوئیم الگوریتم بدون انشعاب است.

تصمیم‌گیری و انشعاب

Decision and Branch

- ◆ الگوی «تصمیم‌گیری و انشعاب» مدل کارآمدتری برای پردازش داده‌ها و حل مسأله در قالب در یک الگوریتم واحد است.
- ◆ در الگوی «تصمیم‌گیری و انشعاب»، اجرای الگوریتم بسته به آنکه عبارتی شرطی، ارزش درست یا نادرست به خود بگیرد به دو مسیر منشعب می‌شود.

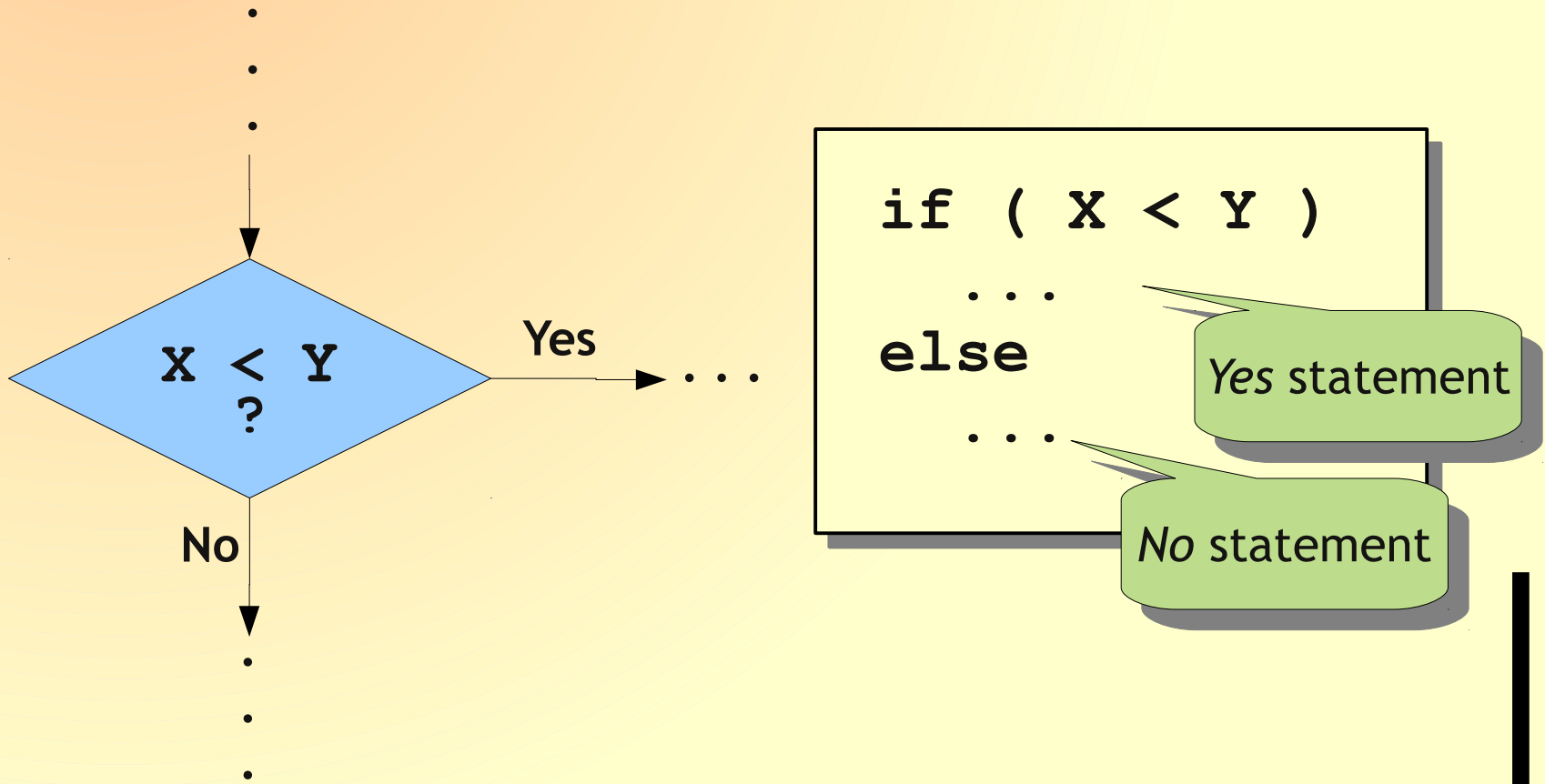
مثال.



الگوریتمی ارائه دهید که
یک عدد صحیح مثبت از
کاربر دریافت کند و تعیین
کند عدد زوج است یا فرد.

تصمیم گیری و انشعاب

Decision and Branch



مثال. الگوریتمی ارائه دهید که یک عدد صحیح مثبت از کاربر دریافت کند و تعیین کند عدد زوج است یا فرد.

```
int main( int argc, char *argv[]){
    int N;
    cin >> N;
    if ( N%2 == 0 )
        cout << N << " is even." << endl;
    else
        cout << N << " is odd." << endl;
    return EXIT_SUCCESS;
}
```

مثال. الگوریتمی ارائه دهید که یک عدد صحیح مثبت از کاربر دریافت کند و تعیین کند عدد زوج است یا فرد.

```
int main( int argc, char *argv[]){
    int N;
    cin >> N;
    if ( N%2 == 0 )
        cout << N << " is even." << endl;
    else
        cout << N << " is odd." << endl;
    return EXIT_SUCCESS;
}
```

عملگرهای حساب

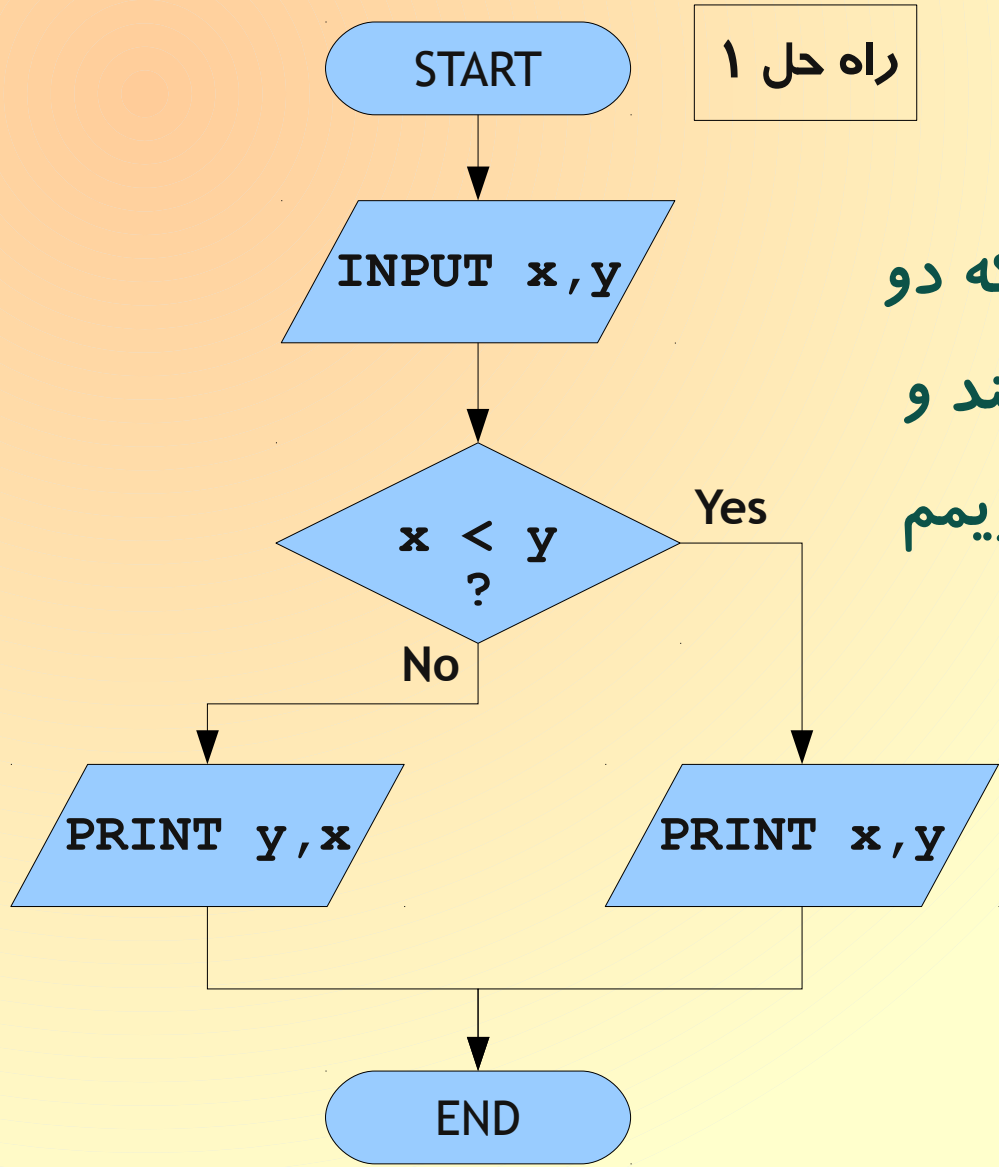
Arithmetic operators

Mathematics	C/C++
+	+
-	-
*	*
/	/
باقیمانده mod	%

عملگرهای مقایسه

Comparative operators

Mathematics	C/C++
=	==
≠	!=
<	<
≥	<=
>	>
≤	>=



راه حل ۱

مثال.

الگوریتمی ارائه دهید که دو عدد از کاربر دریافت کند و به ترتیب مینیمم و ماکزیمم آنها را چاپ کند.

```
#include <iostream>
#include <cstdlib>
using namespace std;

int main( int argc, char *argv[]){
    int x,y;
    cout << "Please, enter two numbers:";
    cin >> x >> y;
    if ( x < y )
        cout << x << "," << y << endl;
    else
        cout << y << "," << x << endl;
    system("pause");
    return EXIT_SUCCESS;
}
```

```

#include <iostream>
#include <cstdlib>
using namespace std;

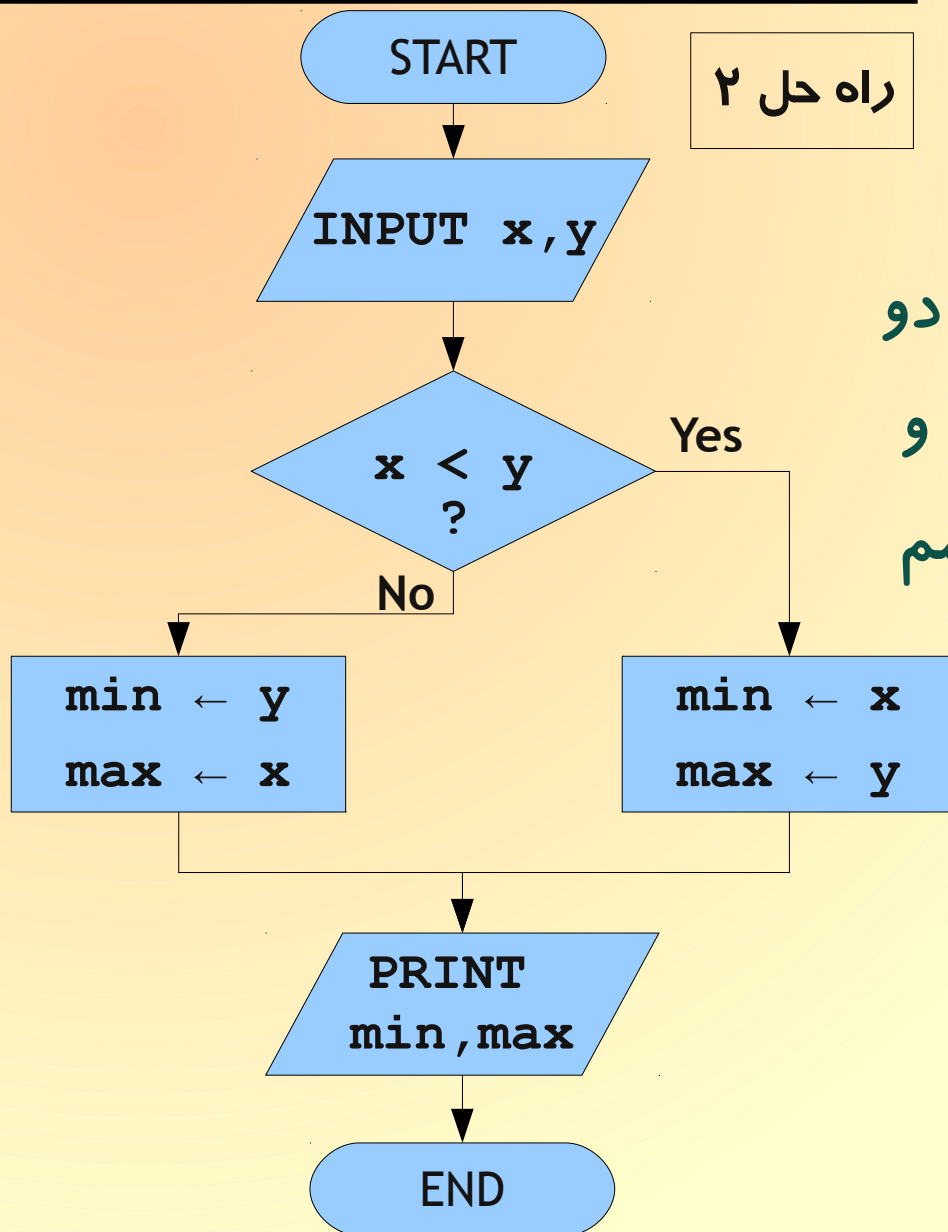
int main( int argc, cha
    int x,y;
    cout << "Please, ente
    cin >> x >> y;
    if ( x < y );
        cout << x << "," << y << endl;
    else;
        cout << y << "," << x << endl;
    system("pause");
    return EXIT_SUCCESS;
}

```

یادآوری. if و else با دستوری که پس از آن‌ها می‌آیند کامل می‌شوند.

بنابراین اگر بلافاصله پس از if یا else علامت ; بگذارید یعنی در برنامه گفته‌اید هیچ عملی در پاسخ به if و یا else انجام نده.

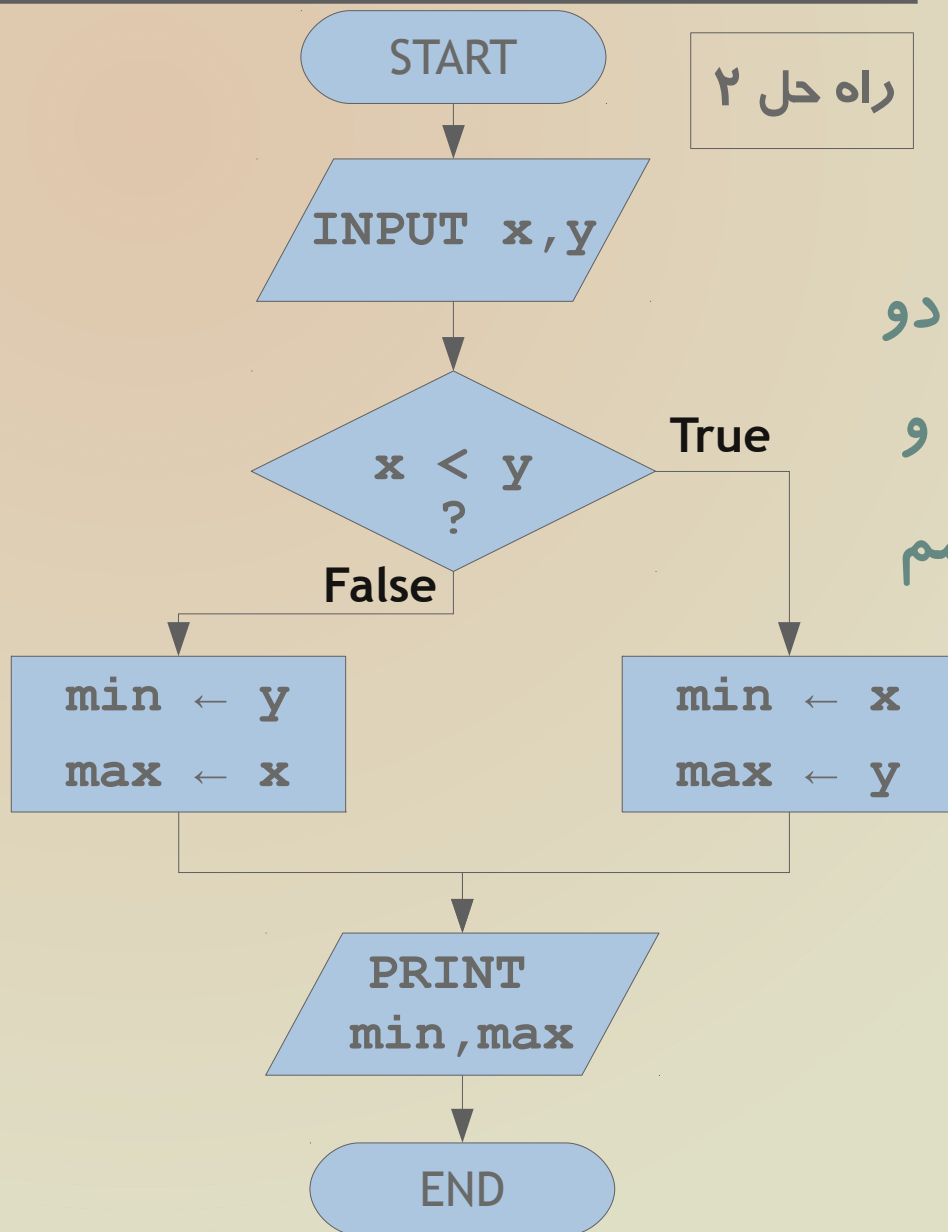
Error!



راه حل ۲

مثال.

الگوریتمی ارائه دهید که دو عدد از کاربر دریافت کند و به ترتیب مینیمم و ماکزیمم آنها را چاپ کند.



راه حل ۲

مثال.

الگوریتمی ارائه دهید که دو عدد از کاربر دریافت کند و به ترتیب مینیمم و ماکزیمم آنها را چاپ کند.

```
int main( int argc, char *argv[]){
    int x,y;
    cout << "Please, enter two numbers:";
    cin >> x >> y;
    int min,max;
    if ( x < y ){
        min = x;
        max = y;
    }
    else {
        min = y;
        max = x;
    }
    cout << min << ", " << max << endl;
    . . . .
```

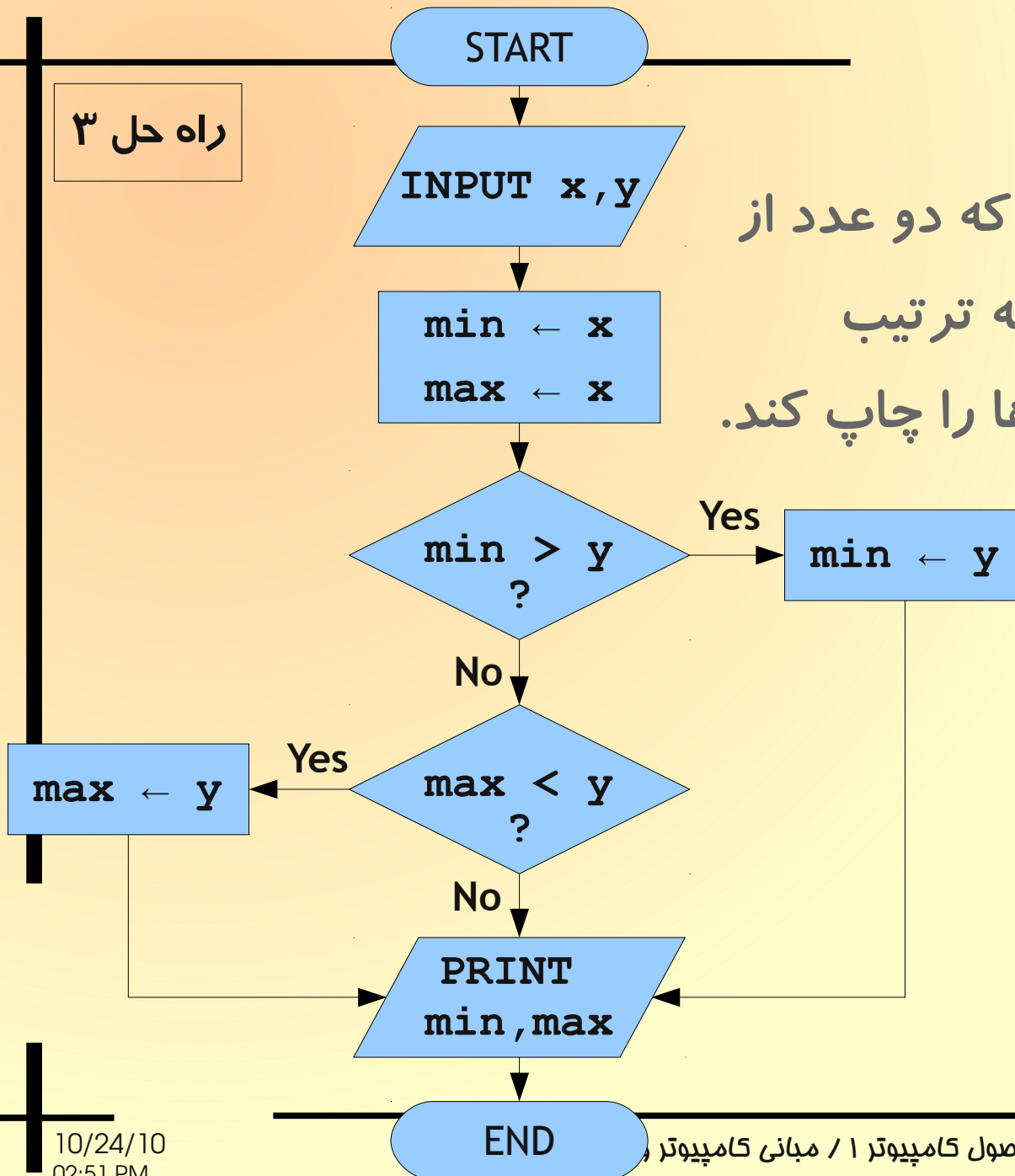
راه حل ۲

بیش از یک دستور در
پاسخ if یا else باید داخل
بلوک { و } قرار گیرد.

مثال.

الگوریتمی ارائه دهید که دو عدد از کاربر دریافت کند و به ترتیب مینیمم و ماکزیمم آنها را چاپ کند.

راه حل ۳




```

int main( int argc, char *argv[]){
    int x,y;
    int min,max;
    cout << "Please, enter two numbers:";
    cin >> x >> y;
    min = x;
    max = x;
    if ( min > y )
        min = y;
    else {
        if ( max < y )
            max = y;
    }
    cout << min << ", " << max << endl;
    . . . .
}

```

راه حل ۳

اگر برای حالت else دستوری
نداریم لازم نیست بنویسیم
else
;

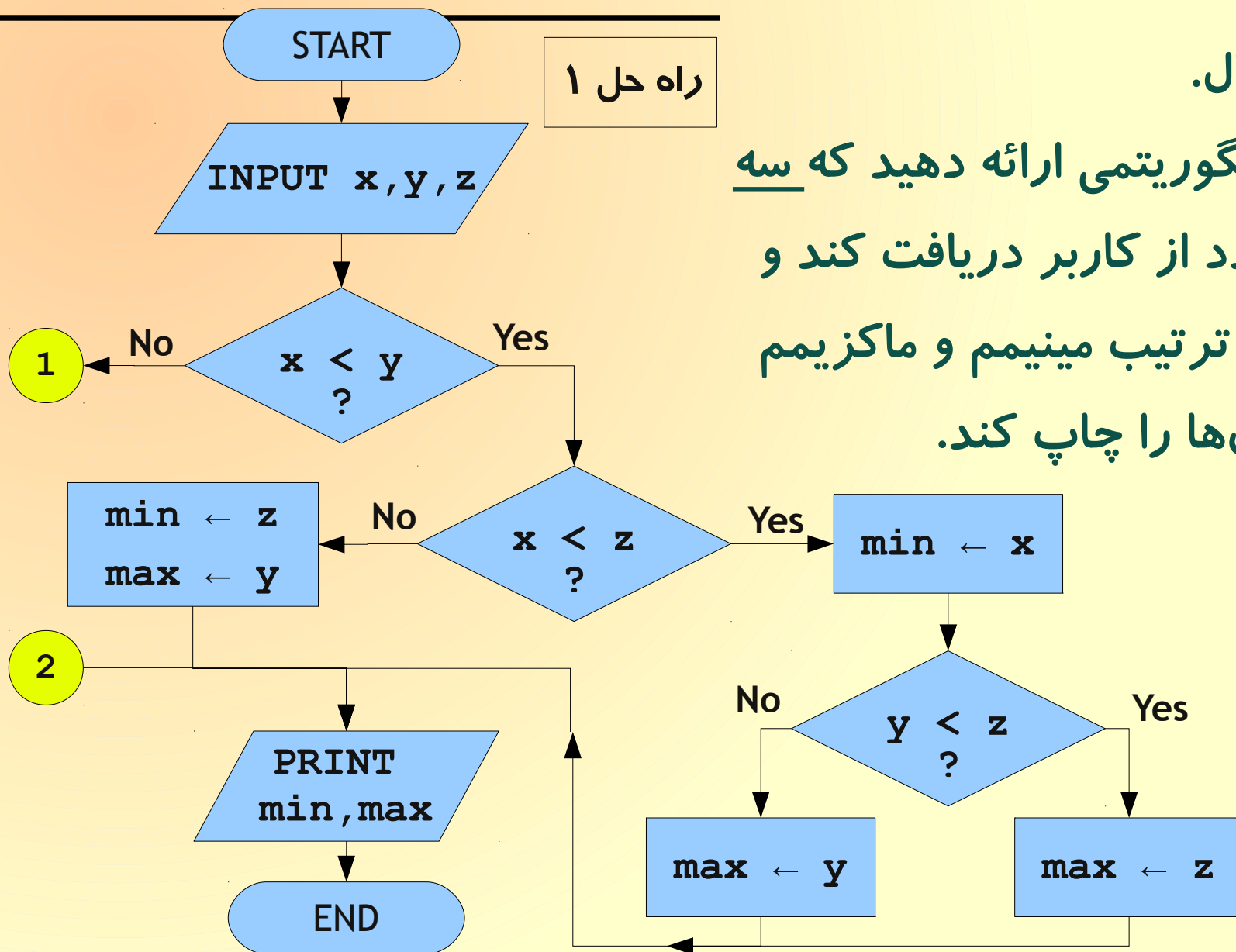
```
int main( int argc, char *argv[]){
    int x,y;
    int min,max;
    cout << "Please, enter two numbers:";
    cin >> x >> y;
    min = x;
    max = x;
    if ( min > y
        min = y;
    else
        if ( max < y )
            max = y;
    cout << min << ", " << max << endl;
    . . . .
```

{ و } را می‌توان ننوشت
چون در نهایت یک دستور
در جواب else آمده است.

مثال.

الگوریتمی ارائه دهید که سه عدد از کاربر دریافت کند و به ترتیب مینیمم و ماکزیمم آنها را چاپ کند.

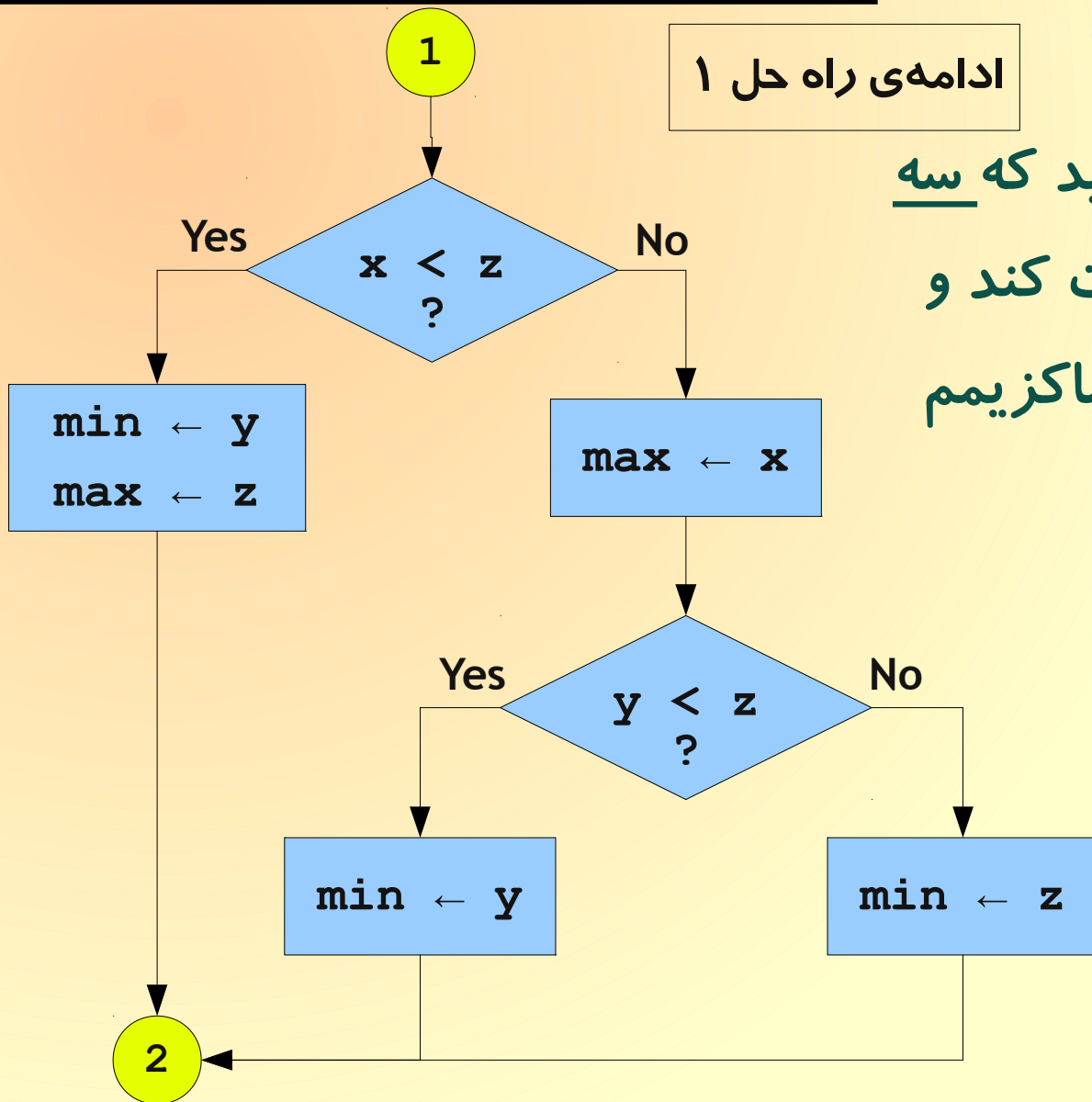
راه حل ۱



مثال.

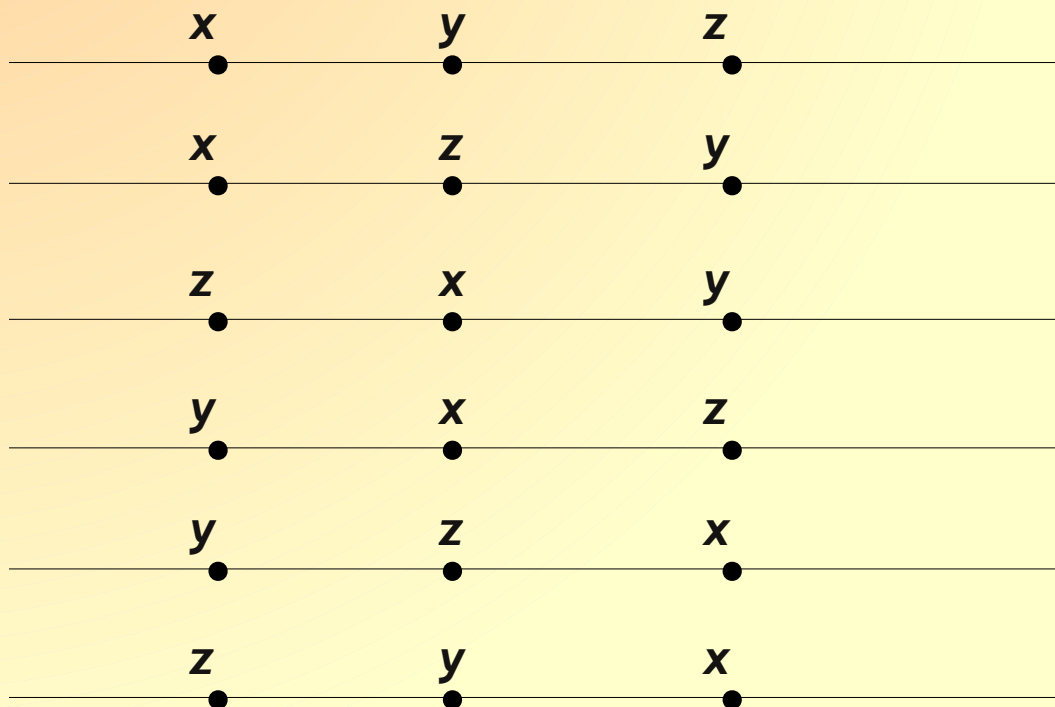
ادامه‌ی راه حل ۱

الگوریتمی ارائه دهید که سه عدد از کاربر دریافت کند و به ترتیب مینیمم و ماکزیمم آنها را چاپ کند.

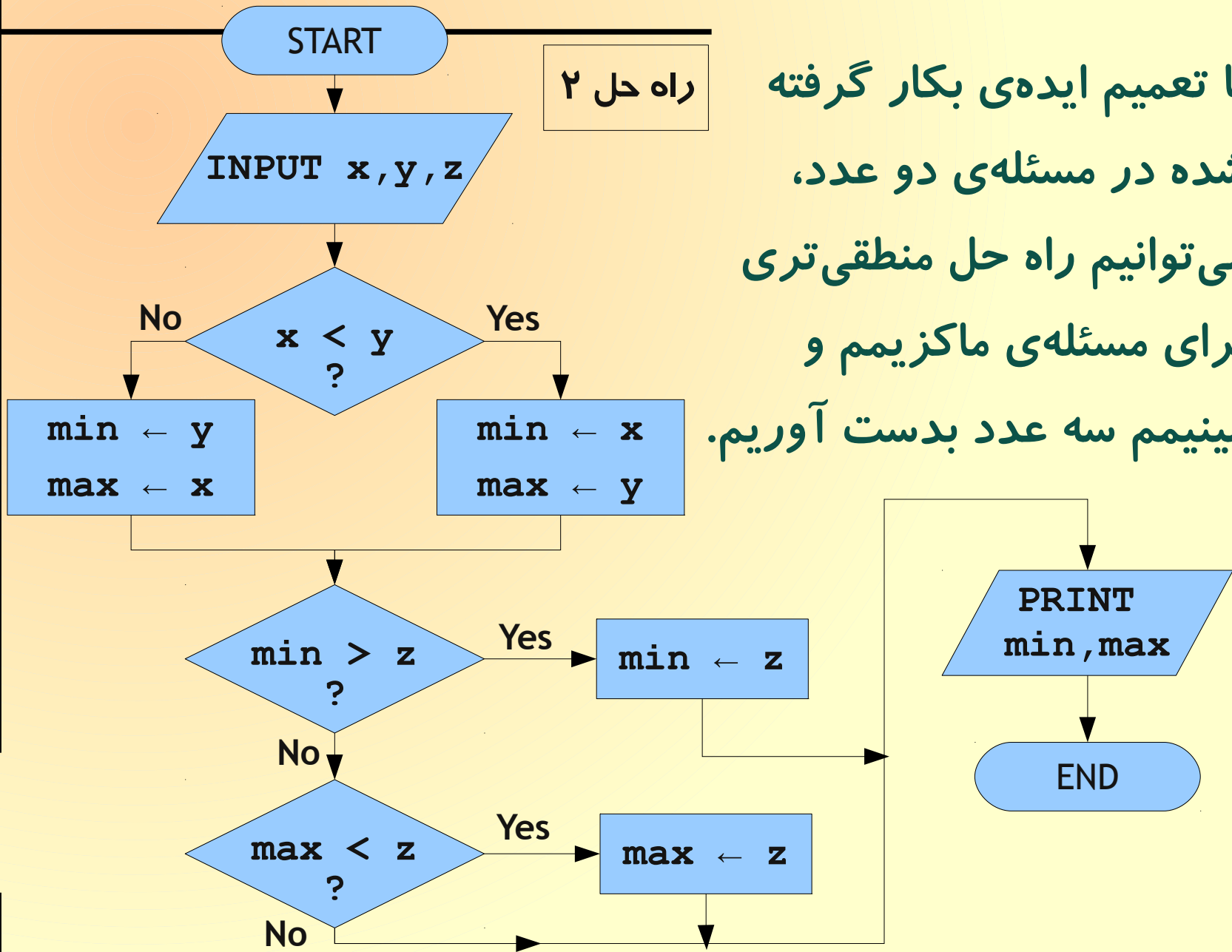


◆ راه حل فوق، در واقع تمام حالت‌هایی بررسی می‌کند که می‌توان سه نقطه را بر روی خط حقیقی نامگذاری کرد به قسمی که نامگذاری‌های متفاوتی برای ماکزیمم و مینیمم بدست آید.

◆ این راه حل از نظر الگوریتمی روش جالبی نیست.



با تعمیم ایده‌ی بکار گرفته شده در مسئله‌ی دو عدد، می‌توانیم راه حل منطقی‌تری برای مسئله‌ی ماکزیمم و مینیمم سه عدد بدست آوریم.



راه حل ۲

```
int main( ){
    int x,y,z;
    cout << "Please,
enter three numbers:";
    cin >> x >> y >> z;
    int min,
    if ( x <
        min =
        max = y,
    }
    else {
        min = y;
        max = x;
    }
}
```

می‌توان { و } را برداشت
چون تنها یک دستور در
پاسخ else اجرا می‌شود.

```
if ( min > z )
    min = z;
else {
    if ( max < z )
        max = z;
}
cout << min << ", "
    << max << endl;
system("pause");
return EXIT_SUCCESS;
}
```

```
int main( ){
    int x,y,z;
    cout << "Please,
enter three numbers:";
    cin >> x >> y >> z;
    int min,max;
    if ( x < y ){
        min = x;
        max = y;
    }
    else {
        min = y;
        max = x;
    }
}
```

```
if ( min > z )
    min = z;
else if ( max < z )
    max = z;
cout << min << ", "
    << max << endl;
system("pause");
return EXIT_SUCCESS;
}
```