



## نکات کلی

- برنامه‌های ارسالی از لحاظ کامپایل شدن نباید با خطا مواجه شوند، در صورت کامپایل نشدن برنامه کل نمره آن مسئله کسر می‌گردد.
- تمامی فایل‌های کد نوشته‌شده را در قالب یک فایل فشرده‌شده **zip** با فرمت **AP\_FinalProject1\_Family\_Name\_[STD\_ID].zip** در سامانه بارگذاری نمایید. به جای مورد درون **[ ]** شماره دانشجویی خود را وارد کنید.
- علاوه بر نسخه **zip**، همه کدها را در قالب یک فایل واحد **cpp** در قسمت نسخه **cpp** تحویل دهید تا کامپایل و اجرا شود.
- جهت خوانایی برنامه، همه فایل‌های نسخه **zip** بایستی به طور دقیق کامنت‌گذاری شوند. قسمتی از نمره به این موضوع اختصاص داده خواهد شد.
- خواهشمند است انجام تمرین به صورت فردی صورت گیرد. در صورت تشخیص تشابه غیرعادی، از هر دو طرف نمره کسر خواهد شد.

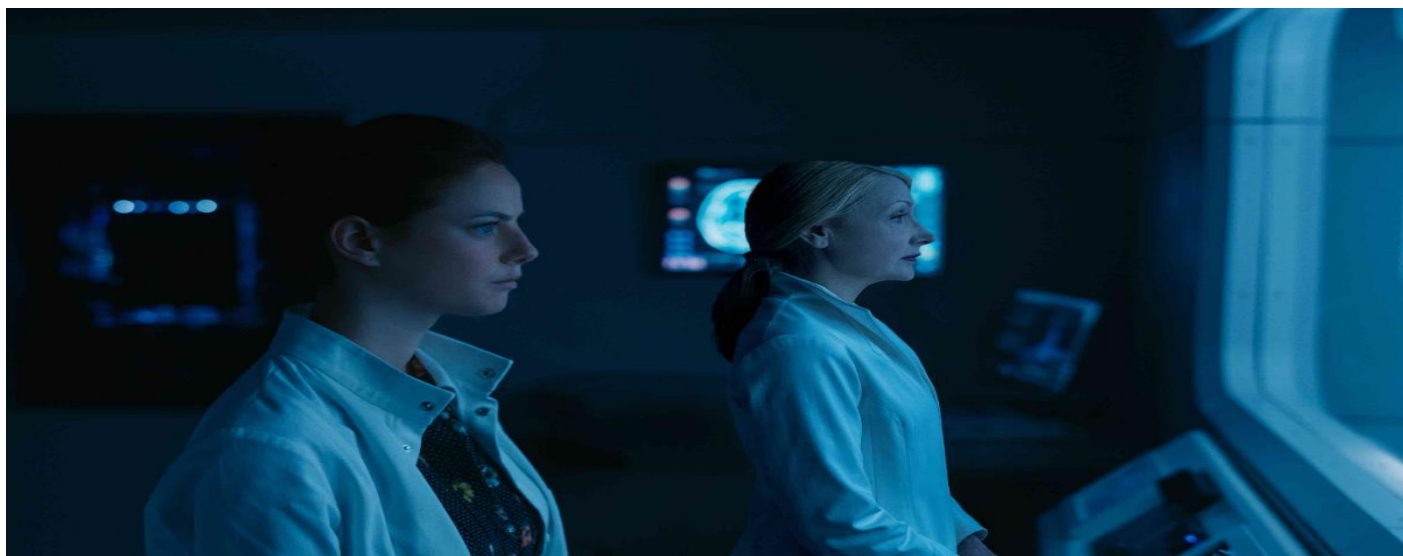
## هزارتو



هنگامی که توماس به همراه گروهی نوجوان در محله‌ای عجیب و غریب و هزارتو از خواب بیدار می‌شوند هیچ‌چیز از گذشته به یاد نمی‌آورند و هیولاهای خشمگین و بی‌رحمی در اطراف آن‌ها پرسه می‌زنند و جان آن‌ها را تهدید می‌کنند. تنها راهی که در مقابل توماس و دیگر نوجوانان گیر افتاده در هزارتو وجود دارد مسیری هزارتو است که اطراف آن را دیوارهای عظیمی در برگرفته است و به ناچار، توماس و بچه‌ها در روشنایی روز از در ورودی وارد هزارتو می‌شوند. اگر زمانی که در تاریکی شب در خروجی هزارتو بسته می‌شود آن‌ها همچنان در هزارتو باشند، شب را در هزارتو دوام نخواهند آورد، چون هیولاهای هزارتو شب‌ها وحشیانه در هزارتو یورش می‌برند و هرچه در سر راه خود ببینند نابود می‌کنند. پس تمام هدف توماس و دوستانش عبور از در خروجی است تا شب را در دام هیولاهای هزارتو نیفتند.

در مسیر رسیدن به در خروجی، اولاً نوجوانان باید سعی کنند مسیری بهینه را طی کنند (چراکه با هر حرکت مقداری از زمانشان کاسته می‌شود). دوماً، ناگزیرند از درهای دیگری نیز عبور کنند که برخی از این درها بدذات و برخی خوش‌ذات هستند. درهای خوش‌ذات بدون هیچ مقاومتی در مقابل توماس یا دیگران به آن‌ها کمک می‌کنند و به اندازه‌ی عدد نوشته‌شده روی در به زمان باقی‌مانده تا بسته شدن در خروجی اضافه می‌کنند و سپس باز می‌شوند. اما درهای بدذات برخلاف درهای خوش‌ذات قصد کمک ندارند و تا معمای موجود روی در حل نشود باز نمی‌شوند. روی هر در بدذات یک نمایشگر و یک صفحه‌کلید تعبیه شده است که نمایشگر معمای در را نشان می‌دهد و نوجوانان برای عبور از آن باید با استفاده از صفحه‌کلید پاسخ صحیح معما را وارد کنند تا در باز شود. در صورتی که پاسخ اشتباه در جواب معما وارد شود آنگاه از میزان زمان باقی‌مانده تا بسته شدن در خروجی به اندازه‌ی عدد نوشته‌شده روی در کاسته می‌شود. به علاوه، صفحه‌کلید موجود روی هر در فقط امکان وارد کردن ارقام و حروف کوچک انگلیسی را فراهم می‌کند و ورود هر کاراکتری غیر از این‌ها پاسخ اشتباه به معما تلقی می‌شود. بعد از اینکه نوجوانان از یک در خوش‌ذات یا بدذات عبور می‌کنند، آن در برای همیشه ناپدید می‌شود.

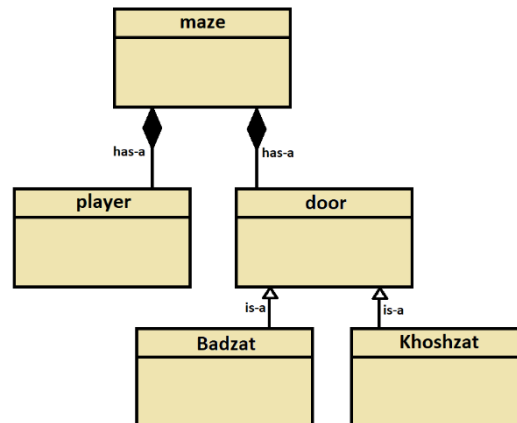
در بیرون از هزارتو، ترپسا دستیار طراح هزارتو یعنی دکتر آوا پیچ، در تلاش است راهی پیدا کند تا بچه‌های گیر افتاده را از هزارتو خارج کند. او یک ایده‌ی محشر برای نجات آن‌ها دارد. اما طبق معمول برای اجرای این ایده، تمام سختی کار به دوش مهندسان نرم‌افزار است. برای نجات بچه‌ها، ترپسا به یک نرم‌افزار شبیه‌ساز از هزارتو نیاز دارد، به همین دلیل ترپسا می‌خواهد طی یک فراخوان، کار ساخت نرم‌افزار شبیه‌ساز را به شما که هم‌اکنون در برنامه‌نویسی ++C به میزان فراوانی مسلط شده‌اید، واگذار نماید.



ترپسا یک شبکه‌ی کامپیوتری فوق سری متصل به هزارتو پیدا کرده و از طریق آن، اطلاعات ساعت مچی‌های توماس و بقیه را در هر لحظه، از طریق این شبکه در اختیار می‌گیرد. اما به دلیل این که او نمی‌داند در خصوص این شبکه‌ی کامپیوتری به چه کسانی می‌تواند اعتماد کند به شما اطلاعات بیشتری در خصوص آن نمی‌دهد و از شما می‌خواهد که علاوه بر نسخه‌ی `cpp` که تحویل می‌دهید و در آن همه‌ی ورودی و خروجی‌ها را از ورودی و خروجی استاندارد خوانده یا چاپ می‌کنید، در نسخه‌ی `zip` تحویلی، ورودی‌ها را از فایل بخوانید و خروجی نهایی را در فایل چاپ کنید تا ترپسا بتواند شخصاً فایل‌هایی را که از شبکه‌ی کامپیوتری به دستش می‌رسد به برنامه‌ی شما بدهد و فایل خروجی را بررسی کند.

ابرقامپیوتر شخصی ترپسا که قرار است برنامه‌ی شما روی آن اجرا شود، ویژگی‌های عجیبی دارد. برای مثال، این ابرکامپیوتر توانایی انجام خود به خودی برنامه‌ها به صورت پردازش موازی را دارد! یعنی حتی اگر شما هم سعی نکنید از این قابلیت‌های پردازش موازی استفاده کنید، کامپایلر اختصاصی ابرکامپیوتر ترپسا، هر دو شی‌ای را که یکی به عنوان ویژگی دیگری ساخته شده باشد، روی دو پردازنده به صورت موازی پردازش می‌کند. قابلیت‌هایی که هنوز علم امروز مهندسی نرم‌افزار در انجام آن عاجز است! البته گفتنی است که ابرکامپیوتر ترپسا اشکالات<sup>1</sup> زیادی هم دارد و ممکن است با توجه به عملکرد و کاربری‌هایی که در اشیاء برنامه وجود دارند، پردازش موازی، نه تنها سرعت اجرا را افزایش ندهد بلکه مقادیر بی‌معنی را اشتباهاً به عنوان خروجی تحویل دهد. اما از آنجایی که آشنایی شما با معماری پردازنده‌ی ابرکامپیوتر ترپسا زمان‌بر است و توماس و بچه‌های گیر افتاده آن قدر زمان ندارند، پس خود ترپسا نمودار ارتباط میان کلاس‌های برنامه‌اش را در اختیار شما قرار داده است تا در نهایت در اجرای برنامه مشکلی به وجود نیاید.

<sup>1</sup> Bug



تریسا از هزارتویی که بچه‌ها در آن گیر افتاده‌اند چیزهای زیادی می‌داند، اما شرط او برای اینکه این داده‌ها را در اختیار شما بگذارد این است که از گذاشتن این اطلاعات در اختیار دیگران اجتناب کنید و هر ویژگی که از هزارتو، افراد گیر افتاده یا هرگونه اطلاعاتی که از تریسا به شما می‌رسد باید در برنامه‌ی شما مخفی شود. همچنین، با توجه به این که تریسا اخیراً به تعدادی از دستیاران دکتر آوا پیچ مظنون شده است و فکر می‌کند آن‌ها مخفیانه در فعالیت‌های او در مورد هزارتو جاسوسی می‌کنند، تمایل دارد که این اطلاعات در پیاده‌سازی مخفی بمانند<sup>۲</sup> و در فایل zip نهایی که به دست او می‌رسد این نکته مورد توجه قرار گرفته باشد. نگرانی‌های امنیتی تریسا به همین‌جا ختم نمی‌شود و او با وجود تمام تمهیداتی که شما برای موارد گفته‌شده در بالا می‌اندیشید معتقد است ممکن است در شرایطی افرادی به برنامه‌ی شما نفوذ کرده و سعی کنند با تغییر دادن برخی مقادیر، کار نرم‌افزار شبیه‌ساز را با مشکل مواجه کنند. به همین دلیل، تریسا به برخی از اطلاعات حساس، برچسب مافوق سری زده است و نرم‌افزار شما باید طوری این اطلاعات را در حافظه‌ی موقت کامپیوتر ذخیره کند که حتی خود برنامه هم در طول اجرای برنامه و پس از مقداردهی اولیه، توانایی تغییر این مقادیر را نداشته باشد. به‌علاوه، برای دسترسی به اطلاعات یک شیء خارج از آن شیء، تنها مجاز هستید به آن شیء دسترسی دوستی<sup>۳</sup> بدهید.

## عملیات‌های مورد نیاز (نسخه **cpp**)

## ۱. ورود اطلاعات اولیه

- نمونه ورودی

1						0
	-	#	#	-	-	
	T		-	#		#
	-	#			T	
	-		-		#	
	S					

## <sup>2</sup> Proxy class

<sup>3</sup> Friend class



1500

3 2

5

2 4

what does the text below means?

death , life

!!!!!!!

life after death

!!!!!!!

4

در خط اول، عدد ۱ که مشخص کننده عملیات ورود اطلاعات اولیه می‌باشد داده می‌شود. در این عملیات، اطلاعات نقشه، میزان کل زمان تا رسیدن شب و جزئیات اطلاعات درها به ترتیب از ورودی گرفته می‌شود. در ابتدا، تریسای اطلاعات نقشه ساده‌سازی شده هزارتو را به نرم‌افزار وارد می‌کند که جزئیات دقیق محل قرارگیری دیوارها، درها و محل ورود و خروج به هزارتو در آن مشخص شده است. در این نقشه، در ورودی به هزارتو در سطر انتهایی و در خروجی هزارتو در سطر اول قرار دارد. حرف O محل در خروجی از هزارتو و حرف S محل در ورودی به هزارتو را مشخص می‌کند. به علاوه، کاراکتر \_ به معنی وجود دیوار افقی، کاراکتر | به معنی وجود دیوار عمودی، کاراکتر # به معنی وجود در بدئات و کاراکتر T مشخص‌کننده‌ی در خوش‌ذات هزارتو است. همچنین، کاراکتر فاصله<sup>4</sup> نشان‌دهنده‌ی فضاهای خالی است. در نمونه ورودی فوق، نقشه هزارتو در ۶ سطر (سطر ۲ الی ۷) داده شده است. پس از نقشه‌ی هزارتو، در سطر ۸ کل زمانی که نوجوانان در اختیار دارند تا از در خروجی خارج شوند، به عنوان ورودی داده می‌شود.

سپس، به تعداد درهای هزارتو که از نقشه قابل استنتاج است، اطلاعات هر در از ورودی گرفته می‌شود. برای ورود اطلاعات هر در، در ابتدا مختصات آن با فرمت [row] [col] از ورودی گرفته می‌شود که در آن بجای [row] و [col] به ترتیب شماره‌ی سطر و ستونی که در، در آن قرار گرفته است وارد می‌شود. دقت کنید که شماره‌گذاری سطر و ستون‌های نقشه از ۱ انجام می‌شود. سپس، در صورتی که آن در خوش‌ذات باشد، شماره‌ی حکاکی شده روی در وارد می‌شود (سطر ۹ الی ۱۰ در نمونه ورودی). این شماره ممکن است برای برخی درها مخدوش باشد و به صورت ۱۰ کاراکتر # پشت سر هم بجای یک عدد صحیح داده شود که در این صورت عدد حکاکی شده روی در با مقدار فوق سری DEFAULT\_T=3 در نظر گرفته می‌شود. چنانچه مختصات داده شده مربوط به یک در بدئات باشد، اطلاعات مربوطه در سه بخش وارد می‌شود که هر بخش آن با یک خط حاوی ۱۰ کاراکتر ! پشت سر هم از دیگر بخش‌ها جدا شده است (سطر ۱۱ الی ۱۷). در بخش اول، صورت سؤال معمای روی در وارد می‌شود و در بخش دوم پاسخ سؤال معما آمده است. بخش اول و دوم ممکن است شامل یک یا چند سطر باشد. در بخش سوم، مقدار عددی که روی در نوشته شده است به صورت یک عدد صحیح مثبت وارد می‌شود. ممکن است در بخش سوم بجای یک عدد صحیح، ۱۰ کاراکتر # پشت سرهم وارد شود که در این صورت باید مقدار فوق سری DEFAULT\_DEC=2 به عنوان عدد روی در لحاظ شود. در نمونه بالا صرفاً جهت مثال اطلاعات یک در خوش‌ذات و یک در بدئات وارد شده است، اما در واقعیت، برنامه تا وقتی که اطلاعات همه درها از ورودی گرفته شود منتظر می‌ماند.

#### • نمونه خروجی در صورت موفقیت

Successful Initialization!

```
      0
  _ _ _ _
| # | # | _ | |
| | _ | # | # |
| _ _ | _ T | |
```

<sup>4</sup> Space



```
| _ | _ | _ | # |
| P _ _ _ _ _ |
Remaining time: 1500 seconds
Enter U to go up, Enter R to go right, Enter Q to terminate the program
```

در صورتی که ورودی‌های فوق به درستی وارد شده باشند، نمونه خروجی بالا که شامل پیام موفقیت عملیات ورود اطلاعات اولیه، نقشه هزارتو، زمان باقیمانده و منوی راهنمای وارد کردن حرکات نوجوانان است چاپ می‌شود. دقت کنید بین خطوط فاصله‌ای وجود ندارد.

#### • نمونه خروجی در صورت عدم موفقیت

Unsuccessful Initialization!

در صورتی که ورودی‌ها اشتباه داشته باشند، برای مثال عدد حکاکی‌شده روی یک در خوش‌ذات یا بدذات نه عدد صحیح باشد و نه ۱۰ کاراکتر! متوالی، خروجی فوق چاپ می‌گردد.

#### ۲. عملیات تلاش برای خروج از هزارتو

##### • نمونه ورودی

```
2
R
```

در خط اول، عملیات ۲ که مربوط به حرکت نوجوانان در هزارتو می‌باشد، آورده شده است. در خط دوم، جهت حرکت نوجوانان نسبت به جایگاه فعلی با کاراکترهای R (راست)، L (چپ)، U (بالا)، D (پایین) مشخص می‌شود. در صورتی که کاربر از ادامه شبیه‌سازی منصرف شود، در خط دوم کاراکتر Q را وارد می‌کند.

#### • نمونه خروجی در صورت رسیدن به یک در خوش‌ذات در اثر حرکت

```
Increased time: 51 seconds
      0
| _ _ _ _ _ |
| # | # | _ _ |
| P | _ _ | # | _ _ |
| _ _ _ _ _ |
| _ _ _ _ _ |
| _ _ _ _ _ |
| S _ _ _ _ _ |
Remaining time: 1551 seconds
Enter U to go up, Enter D to go down, Enter Q to terminate the program
```

در صورتی که ورودی صحیح باشد و در اثر حرکت، نوجوانان به یک در خوش‌ذات برسند، مقدار اضافه‌شده به زمان باقیمانده در خط اول خروجی چاپ می‌شود. در خطوط بعد، نقشه هزارتو چاپ می‌شود. دقت کنید که در نقشه چاپ شده نوجوانان با علامت P در محل در خوش‌ذات و به جای آن نمایش داده می‌شوند و در خوش‌ذات برای همیشه از نقشه حذف می‌شود. در خط هشتم، زمان باقیمانده و در خط نهم، جهت‌های حرکت قابل انتخاب برای حرکت بعدی نوجوانان چاپ می‌شود. برای مثال، چون با توجه به نقشه چاپ شده در این خروجی، نوجوانان امکان حرکت به سمت چپ یا راست را به علت وجود دیوار ندارند، تنها جهت‌های حرکت بالا و پایین به عنوان جهت‌های حرکت قابل انتخاب معرفی شده‌اند.



- نمونه خروجی در صورت رسیدن به یک در بدذات در اثر حرکت

what do you see?

a  
n  
g  
l  
e  
n  
g  
l  
e

Enter your answer:

در صورتی که ورودی صحیح باشد و در اثر حرکت، نوجوانان به یک در بدذات برسند، سوال روی در مطابق فرمتی که در زمان ورود اطلاعات مربوطه در عملیات ۱ وارد شده است، چاپ می‌شود (سطر ۱ الی ۱۰ در این مثال). سپس، پیامی جهت درخواست وارد کردن پاسخ نوجوانان نمایش داده می‌شود (سطر ۱۱ در این مثال). در ادامه، در صورتی که سوال روی در به درستی پاسخ داده شود، خروجی زیر که شامل نقشه هزارتو، زمان باقیمانده تا بسته شدن در خروجی هزارتو و منوی راهنمای حرکت بعدی نوجوانان می‌باشد، چاپ می‌گردد:

```

      0
  _ _ _ _ _
 | # | # | | |
 | T | _ | # | _ |
 | _ P | _ T |
 | _ | _ | _ | # |
 | S _ _ _ _ |

```

Remaining time: 1487 seconds

Enter L to go left, Enter D to go down, Enter R to go right, Enter Q to terminate the program

در مقابل، در صورتی که سوال روی در بدذات به اشتباه پاسخ داده شود، خروجی زیر که شامل میزان کاسته شده از زمان باقیمانده، نقشه هزارتو، زمان باقیمانده تا بسته شدن در خروجی هزارتو و منوی راهنمای حرکت بعدی نوجوانان است، نمایش داده می‌شود:

Decreased time: 43 seconds

```

      0
  _ _ _ _ _
 | # | # | | |
 | T | _ | # | _ |
 | _ # | _ T |
 | _ P | _ | _ | # |
 | S _ _ _ _ |

```

Remaining time: 1457 seconds

Enter U to go up, Enter L to go left, Enter D to go down, Enter Q to terminate the program



- نمونه خروجی در صورت رسیدن به یک فضای خالی در اثر حرکت:

```

      0
  _ _ _ _ _
  | # | # | _ | |
  | T | _ | # | _ |
  | _ | P | _ | T |
  | _ | _ | _ | # |
  | S | _ | _ | _ |

```

Remaining time: 1450 seconds

Enter L to go left, Enter D to go down, Enter Q to terminate the program

در صورتی که متعاقب حرکت نوجوانان، بازیکن در یک فضای خالی قرار بگیرد، به ترتیب نقشه هزارتو، زمان باقیمانده و منوی راهنمای حرکت بعدی نوجوانان در خروجی چاپ می‌گردد.

- نمونه خروجی در صورت اتمام بازی در اثر حرکت:

چنانچه متعاقب حرکت، نوجوانان موفق شوند از هزارتو خارج شوند پیغام برد به صورت رشته حرفی WIN چاپ می‌شود و برنامه با موفقیت پایان می‌یابد. در مقابل، در صورتی که بعد از حرکت نوجوانان در خروجی به علت اتمام زمان بسته شود، پیغام باخت به صورت رشته حرفی LOSE چاپ می‌شود. دقت کنید باخت تنها در صورتی اتفاق می‌افتد که زمان باقی‌مانده تا بسته شدن در خروجی صفر یا مقداری کمتر از صفر شود و تنها در این صورت است که اجرای برنامه متوقف و پیغام باخت باید در خروجی چاپ شود. در صورت اتمام بازی با برد یا باخت متعاقب حرکت نوجوانان، اجرای شبیه‌ساز متوقف می‌شود و در این حالت نقشه، زمان باقی‌مانده و حرکت‌های مجاز برای سیکل بعدی چاپ نمی‌شوند.

- نمونه خروجی در صورت وارد کردن کاراکتر Q در ورودی

Program is terminated!

در صورت درخواست بازیکن جهت خروج از برنامه، پیام فوق چاپ و ادامه شبیه‌سازی متوقف می‌شود.

- نمونه خروجی در صورت عدم موفقیت

Invalid input

در صورتی که ورودی عملیات ۲ اشتباه باشد (ورودی نه کاراکتر Q و نه یکی از جهات حرکت داده شده به عنوان انتخاب‌های ممکن در آخرین خروجی باشد)، پیام بالا در خروجی چاپ می‌شود.



## توضیح ورودی و خروجی نسخه zip

در نسخه zip خواسته شده، همه ورودی و خروجی‌ها مطابق نسخه cpp است. البته برخی از ورودی و خروجی‌ها به جای ورودی و خروجی استاندارد، از فایل گرفته می‌شوند یا روی فایل ذخیره می‌شوند. مشخصات فایل‌های ورودی و خروجی نسخه zip در زیر آورده شده است:

### • فایل‌های ورودی نرم‌افزار

✓ فایل اطلاعات اولیه نقشه هزارتو و زمان کل (maze.txt)

در این فایل، نقشه هزارتو با فرمتی که در نسخه cpp گفته شد و زمان کل تا رسیدن شب آورده شده است و به عنوان ورودی به نرم‌افزار داده می‌شود. به نظر ترپسا این فایل به لحاظ امنیتی فایل خیلی حساسی نیست پس آن را با notepad ایجاد کرده است در نتیجه این فایل از نوع ترتیبی است.

[مثالی از فایل maze.txt برای دانلود](#)

✓ فایل‌های اطلاعات درهای هزارتو به تفکیک

به ازای هر در موجود در هزارتو، یک فایل با نام [row]\_[col].txt در مسیر doors\ به نرم‌افزار داده می‌شود که در نام فایل بجای [row] و [col] به ترتیب شماره‌ی سطر و ستون مربوط به در، منطبق بر فایل maze.txt جایگزین شده است. در فایل مربوط به یک در خاص، مشخصات مربوط به آن در خوش‌ذات یا بدذات مطابق قالب گفته‌شده در توضیح نسخه‌ی cpp آورده شده است. ملاحظات امنیتی این فایل‌ها مشابه فایل maze.txt می‌باشد.

[مثالی از پوشه‌ی doors برای دانلود](#) (مطابق با فایل maze.txt داده‌شده در بالا)

### • فایل خروجی نرم‌افزار

در نسخه zip، خروجی برنامه به صورت پیغام برد (رشته‌ی حرفی WIN) یا باخت (رشته‌ی حرفی LOSE) به جای خروجی استاندارد در فایل result.out ذخیره می‌شود. فایل خروجی بایستی به گونه‌ای ذخیره‌سازی شود که اگر این فایل توسط یک عامل نفوذی به هر کامپیوتر دیگری با معماری متفاوت ذخیره‌سازی داده‌ها منتقل گشت، اطلاعات ذخیره‌شده در آن بی‌معنی باشند. در این مورد ترپسا به شما اطمینان می‌دهد که معماری ذخیره‌سازی اطلاعات در ابرکامپیوترش با هیچ کامپیوتر دیگری در جهان یکی نیست.

## نمونه ورودی و خروجی

جهت خودسنجی آسان‌تر برنامه‌ها، دو نمونه ورودی و خروجی<sup>۵</sup> متناظر در زیر قابل بارگیری است. دقت کنید به پاس کردن این نمونه ورودی و خروجی‌ها که تحت عنوان تست کیس‌های ۱ و ۲ نیز در کوئرا آپلود شده‌اند، نمره‌ای تعلق نمی‌گیرد و فقط جهت راهنمایی و سنجش ایرادات برنامه در اختارتان قرار گرفته است.

[دانلود نمونه ورودی ۱](#)

[دانلود نمونه خروجی ۱](#)

<sup>5</sup> Test-case





[دانلود نمونه ورودی ۲](#)

[دانلود نمونه خروجی ۲](#)

### نکات

- مکانی که بازیکن در آن قرار دارد با کاراکتر P نمایش داده می‌شود.
- وقتی که دری اعم از خوش‌ذات یا بدذات در هزارتو باز شود، در سیکلی که نوجوانان در آن موقعیت در حال باز کردن در است، در آن محل کاراکتر P چاپ می‌شود. در سیکل‌های بعد، بجای در کاراکتر فاصله قرار می‌گیرد.
- در صورتی که بازیکن به سوال یک در بدذات پاسخ نادرست بدهد، کاراکتر P در مکان قبلی‌اش نمایش داده می‌شود یعنی در باز نشده و بازیکن همچنان پشت در قرار دارد.
- اگر نمایش کاراکتر P در یک موقعیت مکانی با وجود یک دیوار افقی تداخل داشت موقتاً در آن مکان کاراکتر P نمایش داده شده و کاراکتر \_ (دیوار افقی) نمایش داده نمی‌شود. البته از سیکل بعدی که نوجوانان دیگر در آن موقعیت حضور ندارند، کاراکتر \_ دوباره نمایش داده می‌شود. به‌عنوان مثال، هزارتوی زیر را قبل و بعد از انجام حرکت به سمت راست مشاهده نمایید.

```
  O
 | P |
 | S |
```

قبل از حرکت به راست

```
  O
 | P |
 | S |
```

بعد از حرکت به راست

- افزوده‌شدن زمان به زمان باقی‌مانده تا بسته شدن در خروجی در اثر مواجهه با یک در خوش‌ذات و کم شدن زمان به دلیل پاسخ نادرست به معمای یک در بدذات، باید به ترتیب به کمک بازتعریف عملگرهای + و - پیاده‌سازی شوند.
- ترتیب نمایش جهت حرکات ممکن در عملیات ۲ باید از سمت چپ مطابق ترتیب (U,L,D,R,Q) باشد، برای مثال، در پیام

Enter L to go up, Enter U to go left, Enter Q to terminate the program

- ترتیب داده شده اشتباه است، چون جهت حرکت چپ قبل از جهت حرکت بالا آمده است
- مقدار زمانی که هر حرکت بازیکن در یک سیکل بازی، از زمان بازیکن کسر می‌کند ۱۰ ثانیه است.