

Welcome Interns!

Welcome to the team! We have designed a series of tasks to help you improve your logical thinking and coding skills. Each task is aimed at enhancing your ability to solve problems programmatically. The tasks cover a variety of challenges, from basic algorithm development to more complex problem-solving. We encourage you to think creatively and explore different approaches to these problems. Good luck and happy coding!

Task 1: Reverse String Without Affecting Special Characters

Reverse the string without affecting special characters and numbers.

Examples

```
revSpecstring("AFC#47GH$Ieu") → "ueI#47HG$CFA"  
revSpecstring("guyhiuj1234!@#$$%rtyhghu") → "uhghytr1234!@#$$%juihyug"  
revSpecstring("12!@" ) → "12!@"
```

Notes

Try with for loops.

Task 2: Prince of Persia Turned Around

You are working for Jaffar in the newest game of Prince of Persia. The prince is coming to get the princess and you have to stop him. He is entering the castle on a horse, do not ask me why he is riding a horse... he just is!

You are standing next to the cannon and you have to check if the aim / velocity / height is ok for hitting the prince on his horse.

Your function will get four values / circumstances:

1. Velocity
2. Angle
3. Height
4. Distance to the prince

With the formula of **Ballistic Trajectory** you will be able to calculate the distance the cannonball is gonna travel for impact. You do not need to apply rounding.

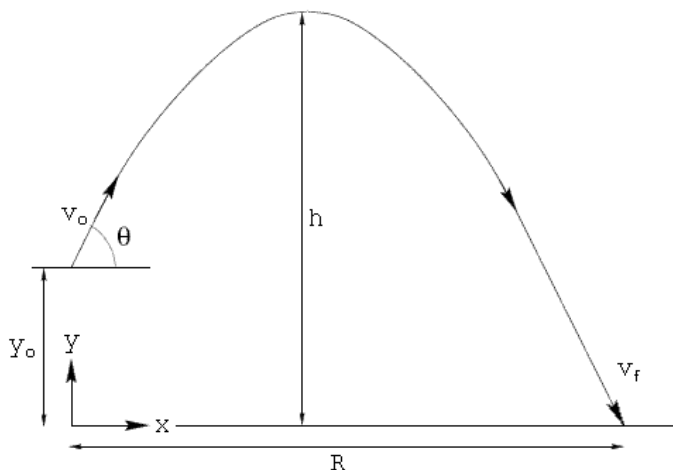
The complete formula is found online. Computations are based on the acceleration of gravity on the earth's surface (9.81 m/s/s), atmospheric drag is neglected. The chance of hitting the prince / his horse is plus or minus 0.5m.

Examples

```
hitPrince(10, 10, 10, 16) → true
```

```
hitPrince(20, 45, 0, 45) → false
```

```
hitPrince(5, 45, 10, 6) → true
```



The parabolic trajectory of a projectile

Notes

- No actual princes / horses are harmed during the making of this challenge.
- All the inputs are correct. $0 > \text{Angle} < 90$, and so on.
- Values will be in meters per second / degrees / meters.

Task 3: Message from Space

You have received an encrypted message from space. Your task is to decrypt the message with the following simple rules:

- Message string will consist of capital letters, numbers, and brackets only.
- When there is a block of code inside the brackets, such as [10AB], it means you need to repeat the letters **AB** for **10 times**.
- Message can be embedded in multiple layers of blocks.
- Final decrypted message will only consist of capital letters.

Create a function that takes encrypted message string and returns the decrypted message.

Examples

```
spaceMessage("ABCD") → "ABCD"
spaceMessage("AB[3CD]") → "ABCD CDCD"
// "AB" = "AB"
// "[3CD]" = "CDCD"
spaceMessage("IF[2E]LG[5O]D") → "IFEELGOOOOOD"
```

Task 4: Helping Alex with Treasure

Alex and Cindy, two students who recently spent some time on treasure hunting. Apart from scrap metal, they found a number of boxes full of old coins. Boxes are of different value and now are lined up in a row. Cindy proposes an idea to divide the treasure into two parts. She thinks that a fair way is that she and Alex take turns, and each of them chooses one box from either left or right side of the line. Cindy is a very generous person and lets Alex to go first.

Alex wants to check whether this idea is actually good for him. He asks you to write a program to calculate the total value that he will get compared to how much Cindy will get if he chooses a box first. You can be sure that they both are very smart, and always select the next box in such way that it leads to the best overall individual solution for them. This means they may not always choose the highest value box of the two currently available in order to ensure they get a higher value box later.

Examples

```
Solve([7, 2]) → 5
```

```
// Alex will choose the 7, and then Cindy gets the 2.
```

```
// So the result is  $7 - 2 = 5$ .
```

```
Solve([2, 7, 3]) → -2
```

```
// It doesn't matter whether Alex chooses the 2 or the 3. Cindy will
```

```
// choose the 7 and Alex will get the remaining box.  $(2+3) - 7 = -2$ .
```

```
Solve([1000, 1000, 1000, 1000, 1000]) → 1000
```

```
// Since Alex chooses first, he will get 3 boxes and Cindy will get only 2.
```

```
// They all have the same value so  $(1000+1000+1000) - (1000+1000) = 1000$ .
```

```
Solve([823, 912, 345, 100000, 867, 222, 991, 3, 40000]) → -58111
```

Task 5: Crypto Wars: Missing Number

Our fleet managed to get one of the enemy's top-secret codes from the remains of its fallen ship. The codes were immediately sent over to our code-breaking base over at **Bleckley Park** ;) for analysis. The team found that each code contains 25 numbers with one missing. The missing number corresponds to a letter in the English alphabet. Your job is to find a more efficient Method of decrypting the messages by digitizing the process.

Write a function that takes an array, detects the missing number (in the array), and returns its **corresponding letter**.

Examples

```
decrypt([19, 12, 14, 21, 22, 3, 11, 20, 9, 16, 24, 17, 2, 10, 13, 18, 7, 8, 4, 5, 1, 6, 25, 23, 26]) → "O"
```

```
// The missing number is 15.
```

```
decrypt([24, 12, 2, ..., 25]) → "N"
```

```
// The missing number is 14.
```

```
decrypt([24, 12, 2, ..., 25]) → "P"
```

```
// The missing number is 16.
```

Notes

- The array will only contain positive integers from 1 to 26 with one missing.
- There will be no duplicate numbers.
- Return the **capital** letter.