



Tyee Programming Competition Club

2023-2024 Parents & Students Info Night

programcomp@tyeeptsa.org



What is this Club About?

- Learn competitive programming
 - Taught by High School Coaches
- Participate in Programming Competitions
 - ACSL
 - USACO
- Meetings: Thursdays at 7pm
 - Virtual
 - Once a month in person

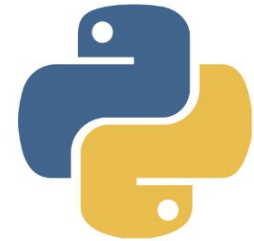
Another Choice: Python Bytes



Tyee PTSA has another programming club:

Python Bytes

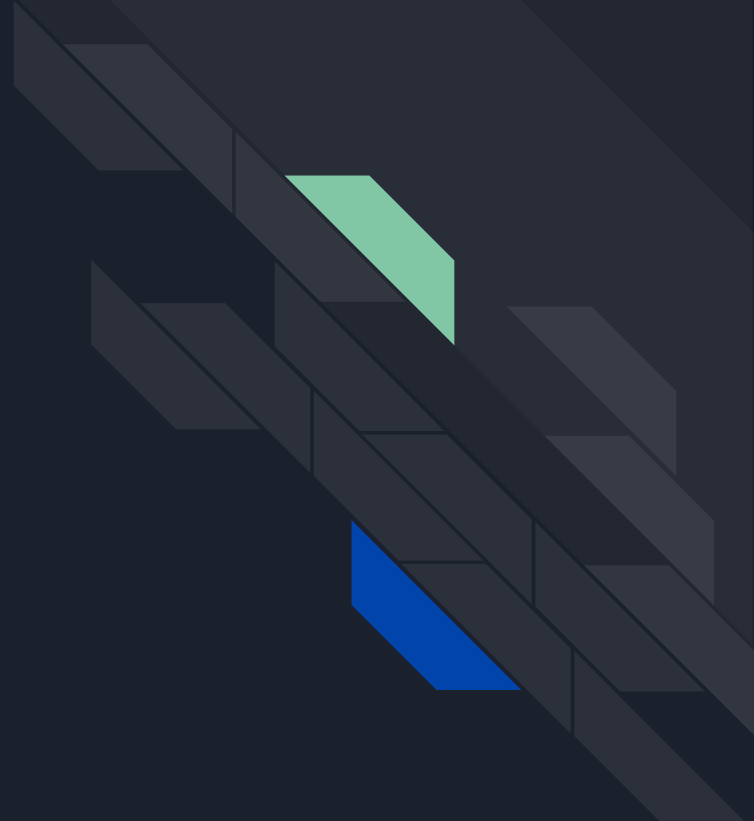
Chair: Reji Dasan



Tyee PTSA Python Bytes Club

People

Chairs and Coaches



Tyee Programming Competition Club Staff

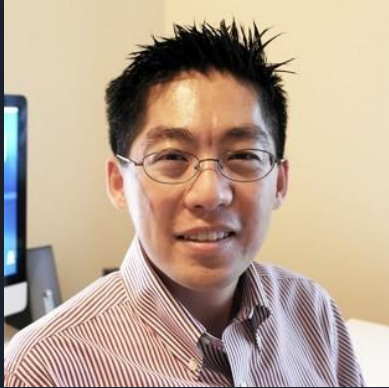


Johannes Grad, PhD (Co-Chair)

- Software Architect at Cadence Design Systems
- Second year of co-chair

Tyee Programming Competition Club Staff

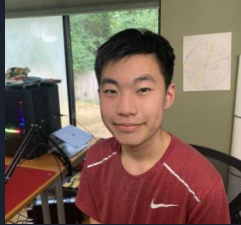
Jason Huh (Adviser)



- Software Engineer at Google
- Founded the club in 2019



Head Coaches - 3 years Experience



Benji Huh

Senior at Interlake High School



Daniel Rhee

Senior at Interlake High School

Senior Coaches - 2 years Experience



Allen Wu
Newport



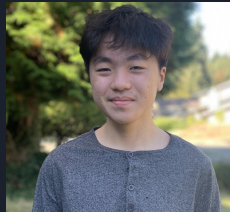
Ethan Do
Newport



Ryan Chen
Newport



Angie Niu
Newport



Lewis Huh
Interlake

Junior Coaches - 1 year Experience



Havish Singavarapu
Newport



Jerryl Tong
University Prep

Assistant Coaches - New this year



Ethan Lee
Interlake



Iliana Rao
Interlake




William Yoon
Interlake

Background

Why Programming Competitions?

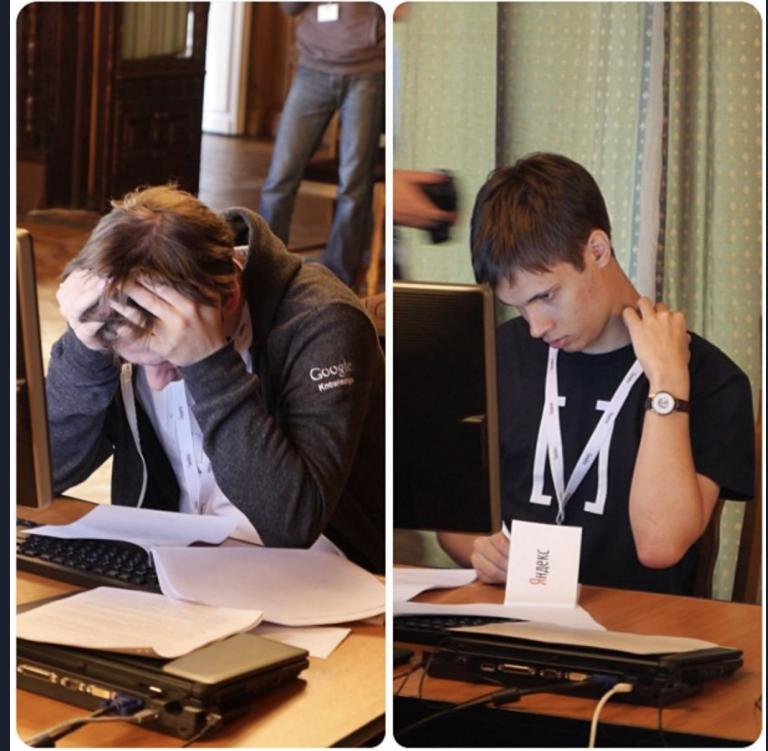




Our mission is to inspire interest in programming and programming competitions by providing a community of like minded students, and to help students advance their knowledge so they can work on projects of their own.

Programming Competitions

- Mental sport
- 1 ~ 4 hours (3 ~ 15 questions)
- Takes many years to master
- Mostly held online

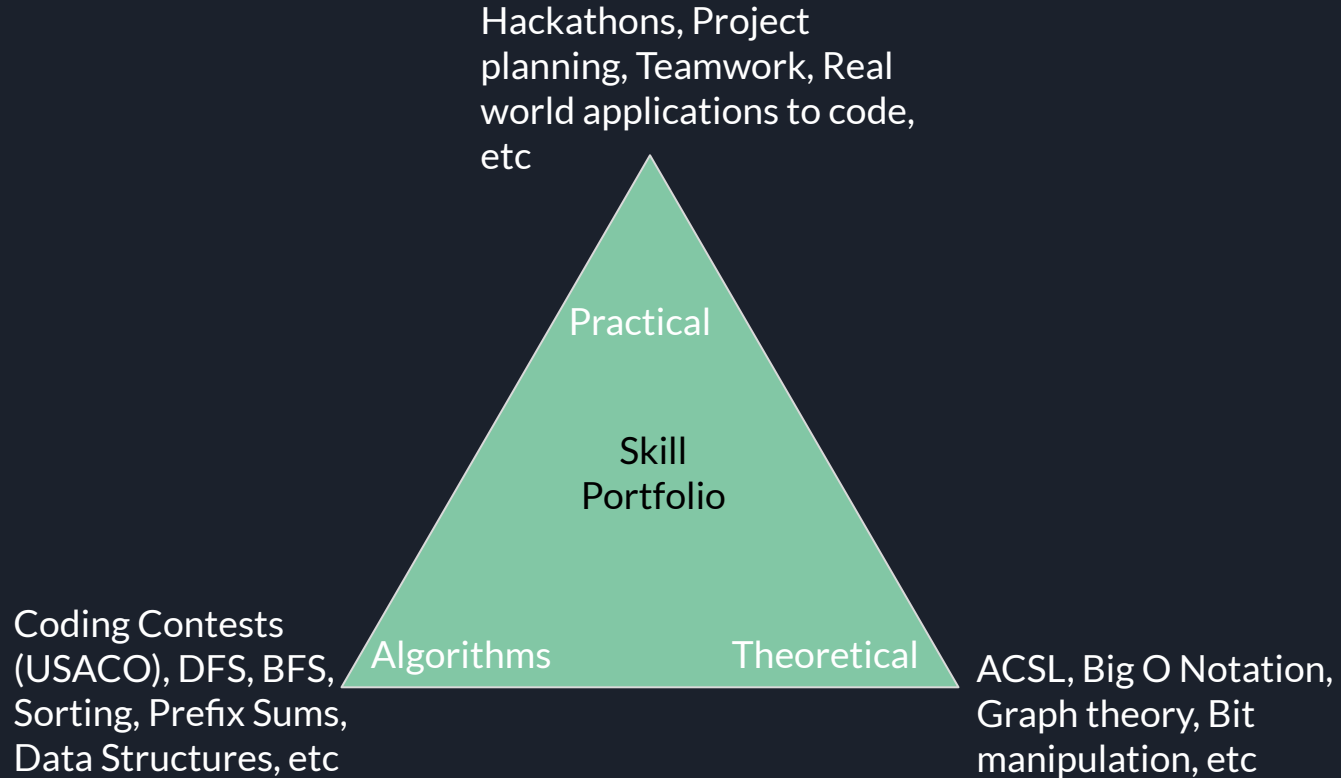


Benefits of Programming Competitions

- Enhances problem-solving skills
- Improve programming skills
- Build solid fundamentals of computer science
- Prepare well for technical interviews
- Learn teamwork
- Intellectually stimulating
(It's fun!)



Student Skills Portfolio





Curriculum

Beginner

- Language Basics (Python, Java)
- Input / Output
- Array / Dynamic Array
- Hash Table
- Binary Search
- Sorting
- Big O Notation
- Digital circuit / Graph
- ACSL prep

Intermediate

- Input / Output
- Array / Dynamic Array
- Hash Table
- Stack / Queue
- Tree
- Big O Notation
- Binary Search
- Sorting
- Digital circuit / Graph
- ACSL prep
- USACO prep (Bronze)

Advanced / Elite

- Array / Dynamic Array
- Hash Table
- Binary Search
- Sorting
- Stack / Queue
- Tree
- Digital circuit / Graph
- ACSL prep
- USACO prep (Bronze)
- USACO prep (Silver)

Programming Competitions that you will join:

- USACO
- ACSL
- TPCC Hackathon

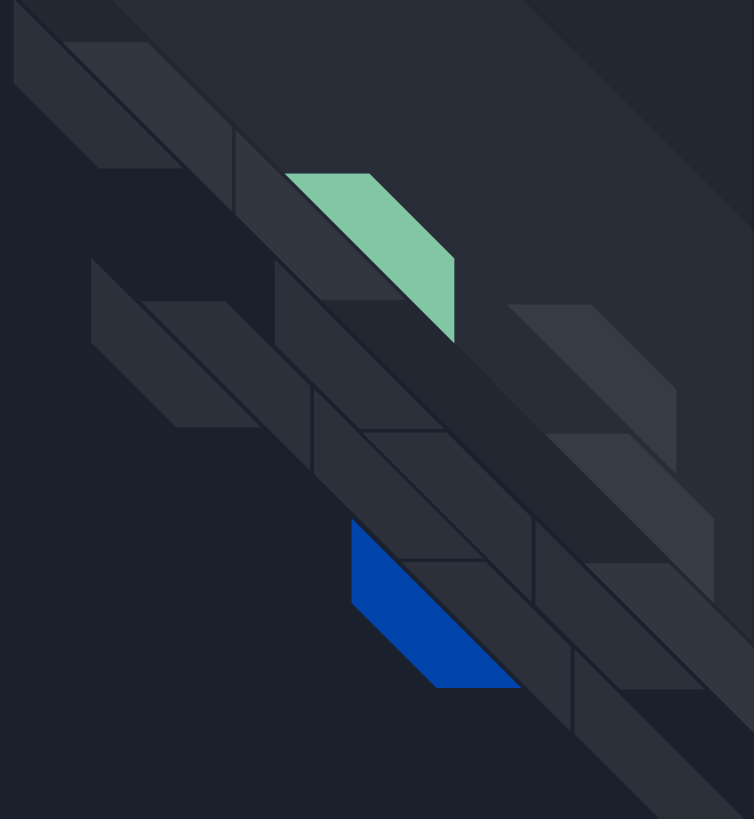


American Computer Science League

ACSL

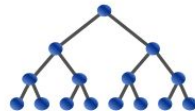
USACO

USA Computing Olympiad



USACO Sample Problem

USA Computing Olympiad

[OVERVIEW](#)[TRAINING](#)[CONTESTS](#)[HISTORY](#)[STAFF](#)[RESOURCES](#)

USACO 2021 DECEMBER CONTEST, PLATINUM PROBLEM 3. HILO

[Return to Problem List](#)

Contest has ended.

Submitted; Results below show the outcome for each judge test case

1 3.3mb 2ms	2 3.3mb 2ms	3 3.3mb 10ms	4 3.3mb 14ms	5 3.3mb 2ms	6 3.3mb 14ms	7 3.3mb 2ms	8 3.3mb 2ms	9 3.3mb 2ms	10 3.3mb 2ms	11 3.4mb 2ms	12 3.4mb 15ms
13 3.5mb 3ms	14 3.6mb 3ms	15 3.8mb 4ms	16 3.9mb 12ms	17 4.1mb 14ms	18 4.3mb 5ms	19 96.4mb 1283ms	20 96.9mb 1915ms	21 97.3mb 1431ms	22 97.6mb 1589ms	23 97.7mb 1198ms	24 98.0mb 975ms
						25 98.2mb 1999ms	26 98.7mb 258ms				

English (en) ▾

Bessie knows a number $x + 0.5$ where x is some integer between 0 to N , inclusive ($1 \leq N \leq 5000$).

Elsie is trying to guess this number. She can ask questions of the form "is i high or low?" for some integer i between 1 and N , inclusive. Bessie responds by saying "HI!" if i is greater than $x + 0.5$, or "LO!" if i is less than $x + 0.5$.

Elsie comes up with the following strategy for guessing Bessie's number. Before making any guesses, she creates a list of N numbers, where every number from 1 to N occurs exactly once (in other words, the list is a permutation of size N). Then, she goes through the list, guessing numbers that appear in the list in order. However, Elsie skips any unnecessary guesses. That is, if Elsie is about to guess some number i and Elsie previously guessed some $j < i$ such that Bessie responded with "HI," Elsie will not guess i and will move on to the next number in the list. Similarly, if she is about to guess some number i and she previously guessed some $j > i$ such that Bessie responded with "LO," Elsie will not guess i and will move on to the next number in the list. It can be proven that using this strategy, Elsie always uniquely determines x regardless of the permutation she creates.

If we concatenate all of Bessie's responses of either "HI" or "LO" into a single string S , the number of times Bessie says "HILO" is the number of length 4 substrings of S that are equal to "HILO."

Bessie knows that Elsie will use this strategy and has already chosen the value of x , but she does not know what permutation Elsie will use. Your goal is to compute the sum of the number of times Bessie says "HILO" over all permutations that Elsie could possibly choose, modulo $10^9 + 7$.

INPUT FORMAT (input arrives from the terminal / stdin):

The only line of input contains N and x .

OUTPUT FORMAT (print output to the terminal / stdout):

The total number of HILOs modulo $10^9 + 7$.

SAMPLE INPUT:

USACO

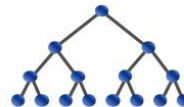
```
-IdeaProjects/competitive-programming/usaco/contest/official/2021-2022/platinum/december — HIL0.cpp [-IdeaProjects/competitive-programming/usaco/contest/official/2021-2022/platinum/december] - NVIM — nvim HIL0.cpp - node
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 #define ll long long
5
6 const int MOD = 1e9 + 7, INV_2 = 500000004;
7
8 int main() {
9     int n, x; cin >> n >> x;
10    vector<vector<int>> nCr(n + 1, vector<int>(n + 1));
11    for (int i = 0; i <= n; ++i) nCr[i][0] = nCr[i][i] = 1;
12    for (int i = 2; i <= n; ++i) for (int j = 1; j < i; ++j) nCr[i][j] = (nCr[i - 1][j - 1] + nCr[i - 1][j]) % MOD;
13    vector<ll> fact(n + 1); fact[0] = 1;
14    for (int i = 1; i <= n; ++i) fact[i] = (fact[i - 1] * i) % MOD;
15    ll ans = 0;
16    for (int i = 1; i <= n; ++i) {
17        int choices = n - x;
18        ll chosen = nCr[choices][i];
19        ll total = (((chosen * fact[i]) % MOD * x) % MOD + fact[n - i - 1]) % MOD;
20        ans = (ans + total) % MOD;
21        for (int l = 1; l < x; ++l) {
22            choices = (1 - l) + (n - x);
23            chosen = nCr[choices][i - 1] - nCr[l - 1][i - 1];
24            total = (((((chosen * fact[i]) % MOD * INV_2) % MOD * (x - l)) % MOD) * fact[n - i - 1]) % MOD;
25            ans = (ans + total) % MOD;
26        }
27    }
28    // Output:
29    cout << ans << endl;
30    return 0;
31 }
```

NORMAL | +0 -0 | master | HIL0.cpp | cpp | utf-8 | 10:53 AM



USACO

USA Computing Olympiad

[OVERVIEW](#)[TRAINING](#)[CONTESTS](#)[HISTORY](#)[STAFF](#)[RESOURCES](#)

PREVIOUS CONTESTS: 2021-2022 SEASON

[2022 US Open Contest Results.](#)[2022 February Contest Results.](#)[2022 January Contest Results.](#)[2021 December Contest Results.](#)

PREVIOUS CONTESTS: 2020-2021 SEASON

[2021 US Open Contest Results.](#)[2021 February Contest Results.](#)[2021 January Contest Results.](#)[2020 December Contest Results.](#)

PREVIOUS CONTESTS: 2019-2020 SEASON

[2020 US Open Contest Results.](#)[2020 February Contest Results.](#)[2020 January Contest Results.](#)[2019 December Contest Results.](#)

PREVIOUS CONTESTS: 2018-2019 SEASON

[2019 US Open Contest Results.](#)[2019 February Contest Results.](#)[2019 January Contest Results.](#)[2018 December Contest Results.](#)

PREVIOUS CONTESTS: 2017-2018 SEASON

[2018 US Open Contest Results.](#)[2018 February Contest Results.](#)[2018 January Contest Results.](#)[2017 December Contest Results.](#)

PREVIOUS CONTESTS: 2016-2017 SEASON

[2017 US Open Contest Results.](#)[Third Contest \(February 2017\) Results.](#)

YOUR ACCOUNT

Welcome, **Allen Wu**

[Edit Account Settings](#)[Logout](#)

2021-2022 SCHEDULE

Dec 17-20: First Contest

Jan 28-31: Second Contest

Feb 25-28: Third Contest

Mar 25-28: US Open

May 26-June 4: Training Camp

Aug 7-14: IOI 2022 in Indonesia



USACO Logistics

- 4 contests per year:
 - December, January, February, March (US Open)
 - Friday-Monday, pick a 4-hour time frame
 - Taken online, link appears at <http://usaco.org/>
- Students need to register on their own
- Scoring
 - 1000/3 points per problem, the number of points you get for each problem is proportional to the number of test cases you get correct per problem (excluding the sample)
 - You are able to see your score immediately after submitting

ACSL

American Computer Science League



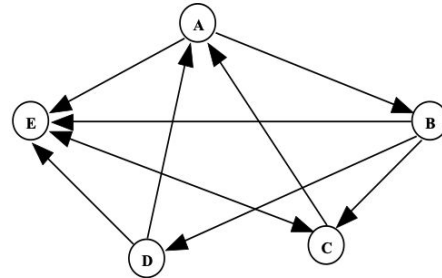
ACSL Sample Problems

Computer Number Systems

Convert $3F6A_{16}$ to octal.

Graph Theory

Draw the adjacency matrix for the directed graph at the right.





ALL	QUESTIONS	TYPE	STATUS	ACTION
①	1. Graph Theory	Multiple Choice	Answered	Modify Submission
1				
2	2. Graph Theory	Multiple Choice	Answered	Modify Submission
3				
4	3. Digital Electronics	Multiple Choice	Answered	Modify Submission
5	4. Digital Electronics	Multiple Choice	Not Answered	Solve Question
	5. Assembly Language	Multiple Choice	Not Answered	Solve Question

[Submit Test](#)



ACSL Logistics

- 4 contests per year
 - December - April
- Parent co-chair Johannes will register all students
- Taken online on the hackerrank platform
 - 5 theory questions
 - 1 programming challenge
- Scoring
 - Up to 10 points per contest
 - Students scoring more than 24 points total will be invited to finals



Last year Tyee team #1 was Second in WA
9 students advanced to finals



Team	ST	C#1	C#2	C#3	C#4	Total
Yuelu Academy #1	WA	47	50	50	42	189
Tyee Middle School PTSA #1	WA	48	50	46	34	178
Yuelu Academy #3	WA	48	49	46	35	178
Yuelu Academy #7	WA	49	49	48	32	178
Yuelu Academy #4	WA	48	50	48	24	170
Yuelu Academy #5	WA	46	49	37	21	153
Redmond Middle School #1	WA	47	45	36	22	150
Tyee Middle School PTSA #2	WA	39	45	40	23	147
Yuelu Academy #2	WA	42	39	35	17	133
Delaware STEAM Academy	DE	39	36	23	23	121
Yuelu Academy #6	WA	38	41	24	9	112
Kirkland Computer Club	WA	22	34	20	22	98
Redmond Middle School #2	WA	10	6	2	1	19

Hackathon

Hack for Good





Hack for Good

[Overview](#)[My projects](#)[Participants \(1\)](#)[Resources](#)[Rules](#)[Project gallery](#)[Updates](#)[Discussions](#)[Manage](#)

Hack for Good (3/19 - 3/24)

Create apps that make the world a better place!

Who can participate

- Above legal age of majority in country of residence - US only
- Middle school students only

[View full rules](#)[View schedule](#)

Opening soon

Online

Public

\$250 in prizes

1 participant

[Tye Middle School Programming Club](#)

Beginner Friendly

Education

Social Good

<https://hackforgoodbsd.devpost.com/>

Hack for Good

[Overview](#)[My projects](#)[Participants \(48\)](#)[Resources](#)[Rules](#)[Project gallery](#)[Updates](#)[Discussions](#)

★ Connect with the participants – support your favorite projects by liking, sharing, and commenting on them.

SORT 

No Virus Game

Oh no! Your computer has been infected with a DEADLY virus! What



♥ 4



A Day in the Life of a Fish

A Day in the Life of a Fish is a scratch project attempting to bring



♥ 0



The Garbage Game

A game about recycling



♥ 1

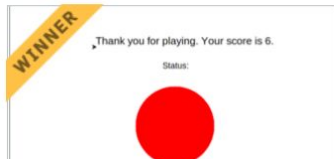
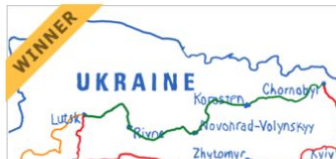


Donate Donut

Come generate your own donuts at Donate Donut with your friends (or



♥ 4



Hackathon FAQ

- Can I create a game?
 - Yes, you can build anything
- How is my submission judged?
 - We will score your submission using following categories:
Originality, Adherence to Theme, Completion, Learning, Design, Technology
- Can I extend a project that I used before the hackathon?
 - No, you need to create a new project with new code

LeetCode

Online Platform for Coding Challenges



Problem List < > 🔍 Premium Register or Sign in

Description Editorial Solutions (25K) Submissions

1. Two Sum Hint

Easy 👍 51.9K 🗨️ 1.7K ☆ 🔄

🔒 Companies

Given an array of integers `nums` and an integer `target`, return *indices of the two numbers such that they add up to `target`*.

You may assume that each input would have **exactly one solution**, and you may not use the *same* element twice.

You can return the answer in any order.

Example 1:

Input: `nums = [2,7,11,15]`, `target = 9`
Output: `[0,1]`
Explanation: Because `nums[0] + nums[1] == 9`, we return `[0, 1]`.

Example 2:

Input: `nums = [3,2,4]`, `target = 6`
Output: `[1,2]`

Example 3:

Input: `nums = [3,3]`, `target = 6`
Output: `[0,1]`

Display a menu

i C++ | • Auto

```
1 class Solution {
2 public:
3     vector<int> twoSum(vector<int>& nums, int target) {
4
5     }
6 };
```

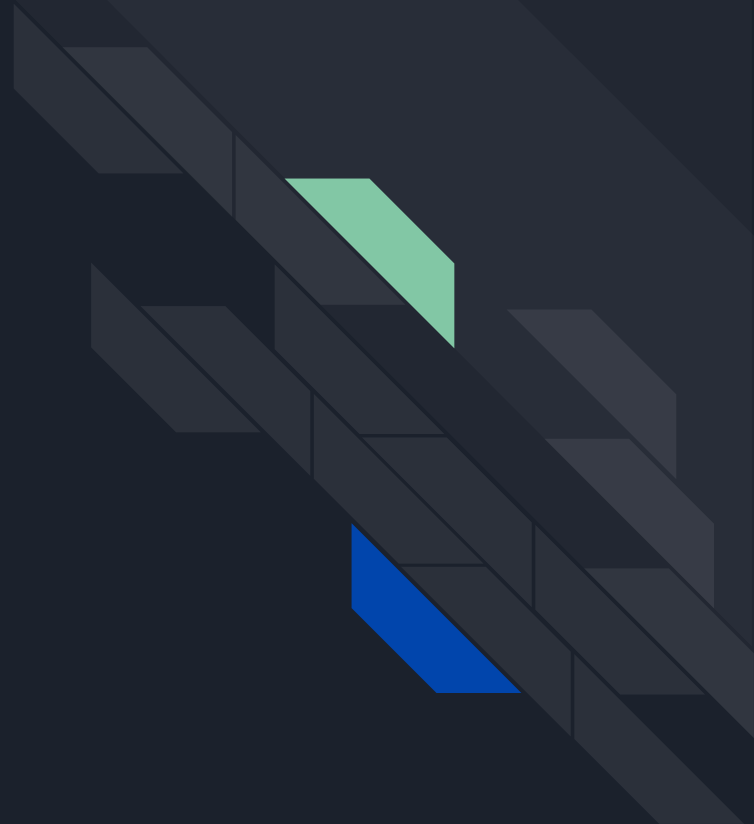
Saved to local

You need to [Login / Sign up](#) to run or submit

Console ^ 🔍 Run Submit

Repl.it

Online IDE



Repl.it

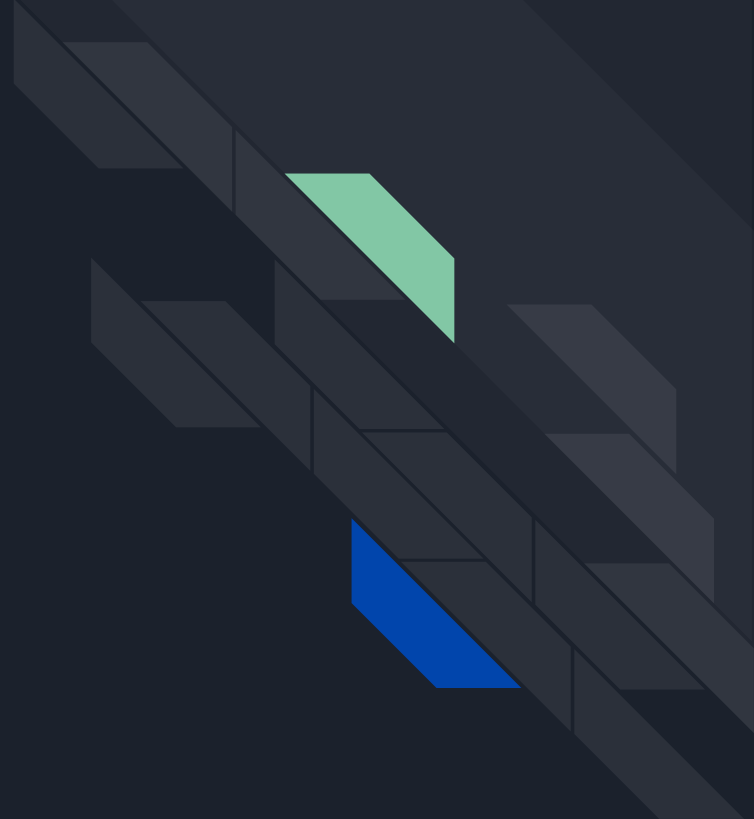
The screenshot displays the Repl.it web interface. At the top, there is a navigation bar with a menu icon, the Repl.it logo, a user profile icon, the text "Replit Intro", a dropdown menu, a "Run" button, and an "Invite" button. Below the navigation bar, the interface is divided into three main sections:

- Files:** A sidebar on the left showing a file explorer with various files like `main.apl`, `main.cpp`, `main.d`, `main.dart`, `main.draw`, `main.go`, `main.java`, `main.md`, `main.php`, `main.scala`, `main.scm`, `main.sh`, and package files like `poetry.lock` and `pyproject.toml`.
- Code Editor:** The central area shows a Python file named `main.py` with the following code:

```
1 #Repl
2 #Sharable with Friends!
3 #Easy to Use
4 #Free to Use
5
6 #Learn Coding With a few easy steps!
7 print("Hello World!") #This says Hello World
8
9 #Make Variables!
10 my_favorite_color = "Pantone 448c"
11 name_of_my_dog = "I don't have one :C"
12
13 #Start Coding a Simple Project
14 class Calculator:
15
16     def addition(a, b):
17         return a + b
18
19     def subtraction(a, b):
20         return a - b
21
22     def multiplacation(a, b):
23         return a * b
24
25     def devision(a, b):
26         if b != 0:
27             return a / b
28         elif b == 0:
29             return None
30
31     def zero_divided_by_zero():
32         #Solution to 0/0
```
- Console:** On the right, a terminal window shows the output of the code execution: `Hello World!`.

At the bottom left, there are system resource indicators for CPU, RAM, and Storage.

Wrap Up





Prerequisites

- Student must have a personal laptop!
 - School laptop blocks most websites
- Some programming in Python, Java or C++



Minimum Bar

Students should be able to solve these two problems:

<https://www.hackerrank.com/challenges/staircase/problem>

<https://www.hackerrank.com/challenges/diagonal-difference/problem>



Club Activities

- Club meetings: Thursdays 7:00 - 8:30 pm
- Hybrid format
 - 1 week per month in-person at Tyee
(Note: probably on Wednesdays)
 - Remaining weeks online via Google Meet
- Online community through Discord



Plan for 2023-2024

- Sep 28, 8pm – Registration opens
- Oct 12, 7-8:00pm – Placement exam
- Oct 19, 7-8:30pm – First weekly club meeting

- Dec - Mar: USACO
- Dec - April: ACSL
- March: 3rd Tyee Hackathon



Registration

- Registration opens *tonight!* (Membership Fee: \$50)
 - <https://tyeepsa.org/Packet/ProgrammingCompetitionReg>
 - Start: 9/28/2023 at 8:00 pm
 - End: 10/11/2023
 - Financial aid available
- Refund Policy
 - Within two weeks from the first meeting
 - Requires Chair's approval



Placement Exam

- Preparation Materials:
 - Practice using Hackerrank: www.hackerrank.com
 - Recommendation:
 - <https://www.hackerrank.com/domains/python>
 - <https://www.hackerrank.com/domains/data-structures>
 - <https://www.hackerrank.com/domains/tutorials/30-days-of-code>
- Placement Exam:
 - Invitation will be emailed after registration closes



PTSA Rules

- Parents need to be PTSA members
- A parent volunteer needs to attend all in-person meetings



Calendar



Tyee PTSA



Membership



Fundraising



Programs



Volunteer

Programming Competition Club 2023-2024



Tyee PTSA
Programming Competition Club

Club Info
Page

Club Chair: *Johnes Grad*

Chess Club
Competition Math Club

Programming Competition

Robotics Club
Science Club
Speech and Debate Club
Spelling Bee

8th Grade Party
Lunar New Year Festival
Reflections Art Program
Student Social Events

Staff Appreciation Events
Walk About Program

Programming Competition Club aims to help students to be proficient in computer programming and computer programming for and participating in various local and online programming competitions using a variety of students' choice (Python, Java or C++).

Students will participate in the [ACSL](#) and [USACO](#) competitions. The club is divided into 4 skill levels, and taught by High School student coaches. Students are expected to have some programming knowledge before joining the club. For beginners, please consider the Python Bytes club instead.

Night

Monday, September 28 at 7pm



Parent Involvement

- Pizza Volunteers
 - Signup Genius will be emailed to parents

- We are looking for more parent co-chairs
 - No experience needed

programcomp@tyeepsa.org



Questions?

Email:
programcomp@tyeepsta.org

