[8 Mar] Operators + Precedence Table

C++, OOPs & CP - Practice Session 2

Special class

[9 Mar] Pointers
10-1r → Practice Questions
↓
① Accuracy
② Speed

Coding Questions

Jay Bansal • Mar 9, 2021

C++, OOPs, Programming

Pointers

Ravindrababu Ravula
Jay Bansal

unacademy.com/@ravula
unacademy.com/@jay.bansal

unacademy | CODECHEF

New Batch Launch

Triumph -

Beginner to Expert Level Coder in 12 Months

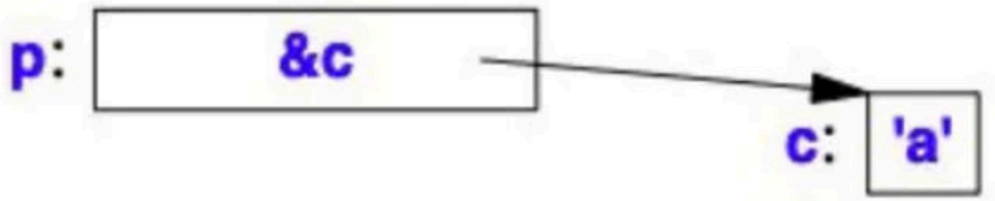
Pointers

For any type **T**, **T*** is the type "**pointer to T**."

That is, a variable of type **T*** can hold the address of an object of type **T**.

For example:

```
char c = 'a';
char* p = &c;  // p holds the address of c; & is the address-of operator
```



Dereferencing or indirection

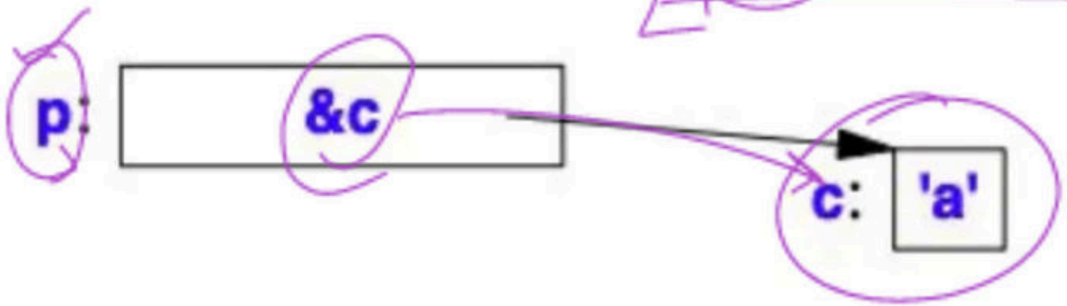
Referring to the object pointed to by the pointer

char c = 'a';

char* p = &c;

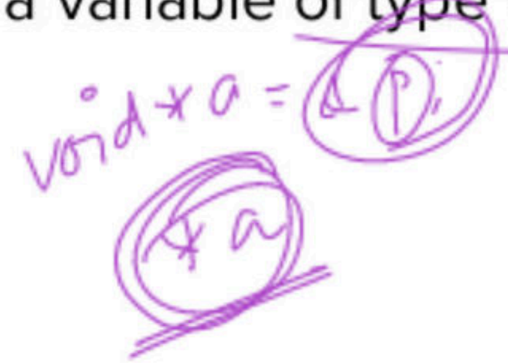
char c2 = *p; // c2 == 'a'; * is the dereference operator

The object pointed to by p is c, and the value stored in c is 'a', so the value of *p assigned to c2 is 'a'



void*

- In low-level code, we occasionally need to store or pass along an address of a memory location without actually knowing what type of object is stored there
- A **void*** is used for that, read as 'pointer to an object of unknown type."
- A pointer to any type of object can be assigned to a variable of type void*
- A void* can be assigned to another void*




```
int* pi;
void* pv = pi; // allowed
```

nullptr

- The literal **nullptr** represents the null pointer, that is, a pointer that does not point to an object

- It can be assigned to any pointer type, but **not** to other built-in types

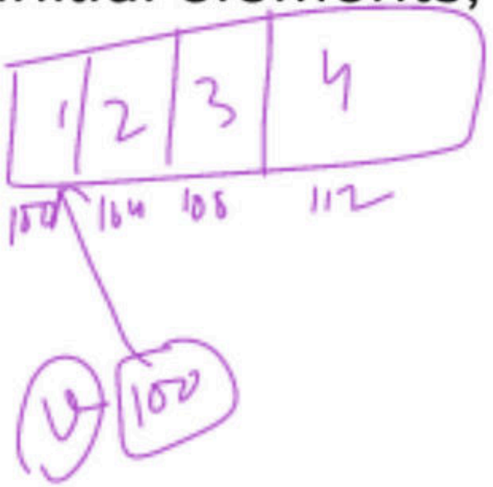
```
int* p = nullptr;
double* p = nullptr;
int p = nullptr; // error: p is not a pointer
```

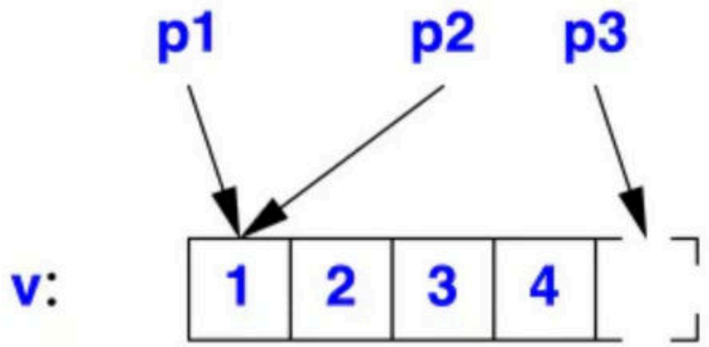
Using nullptr makes code more readable and avoids potential confusion when a function is overloaded to accept either a pointer or an integer (using nullptr instead of 0)

Pointer to Array

The name of an array can be used as a pointer to its initial elements,

```
int v[] = {1, 2, 3, 4};
int* p1 = v;          // pointer to initial element
int* p2 = &v[0];      // pointer to initial element
int* p3 = v+4;
```





int v[4]

v == &v[0]

Pointer to Function

Like a (data) object, the code generated for a function body is placed in memory somewhere, so it has an address

We can have a pointer to a function just as we can have a pointer to an object.

The pointer obtained by taking the address of a function can then be used to call the function

void f(int); foo = &f;

void (*foo)(int); (*foo)(5)

foo is a pointer to a function taking one argument, an integer, and that returns void.

Pointer to Function

void *(*foo)(int *);

Read inside-out

The innermost element of the expression is *foo, and that otherwise it looks like a normal function declaration.

foo is a pointer to a function that returns a void* and takes an int*.

Initializing Pointer to Function

To initialize a function pointer, you must give it the address of a function in your program:

```
void fun(int x) {
    printf( "%d\n", x );
}
void (*ptr)(int);
ptr = &fun;
[or]
ptr = fun;
```

Using Pointer to Function

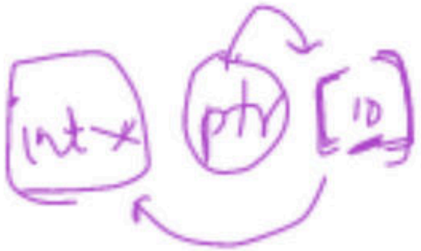
To call the function pointed to by a function pointer, you treat the function pointer as though it were the name of the function you wish to call:

```
void fun(int x) {
    printf( "%d\n", x );
}
void (*ptr)(int);
ptr = &fun;
ptr(5);
[or]
(*ptr)(5);
```

Practice

What does the following declaration means ?

```
int (*ptr)[10];
```



A. ptr is an array of pointers to 10 integers
B. ptr is a pointer to an array of 10 integers
C. ptr is an array of 10 integers
D. ptr is a pointer to an array

Practice

What does the following declaration means ?

```
int (*ptr)[10];
```

A. ptr is an array of pointers to 10 integers

B. ptr is a pointer to an array of 10 integers

C. ptr is an array of 10 integers

D. ptr is a pointer to an array

Practice

Declare the following statement:

"An array of 4 pointers to chars".

A. char *ptr[4]();

B. char *ptr[4];

C. char (*ptr[4])();

D. char **ptr[4];

Practice

Declare the following statement:

"An array of 4 pointers to chars".

A. char *ptr[4]();

B. **char *ptr[4];**

C. char (*ptr[4])();

D. char **ptr[4];

Practice

Declare the following statement?

"A pointer to an array of three chars".

A. char *ptr[3]();
B. char (*ptr)*[3];
C. char (*ptr[3])();
D. char (*ptr)[3];

Practice

Declare the following statement?

"A pointer to an array of three chars".

A. char *ptr[3]();
B. char (*ptr)*[3];
C. char (*ptr[3])();
D. **char (*ptr)[3];**

Practice

```
#include<iostream>
using namespace std;
int fun(int p, int *q, int **r){
    ++*q;
    **r*=2;
    p%=11;
    return (p+*q+**r);
}
int main(){
    int *p, a=5;
    p = &a;
    cout<<fun(a, &a, &p)<<" ";
    cout<<a;
    return 0;
}
```

Output ?

A. Garbage 10

B. 29 12

C. 29 13

D. 29 14

Practice

```
#include<iostream>
using namespace std;
int fun(int p, int *q, int **r){
    ++*q;
    **r*=2;
    p%=11;
    return (p+*q+**r);
}
int main(){
    int *p, a=5;
    p = &a;
    cout<<fun(a, &a, &p)<<" ";
    cout<<a;
    return 0;
}
```

Output ?

A. Garbage 10

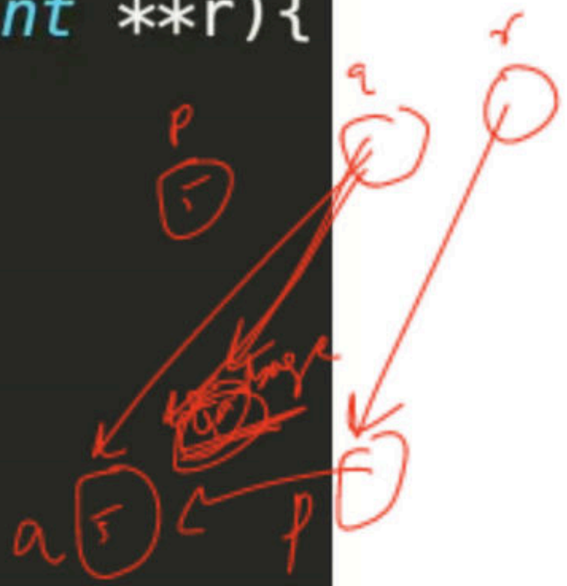
B. 29 12

C. 29 13

D. 29 14

Practice

```
#include<iostream>
using namespace std;
int fun(int p, int *q, int **r){
    *q++;
    **r*=2;
    p%=11;
    return (p+*q+**r);
}
int main(){
    int *p, a=5;
    p = &a;
    cout<<fun(a, &a, &p)<<" ";
    cout<<a;
    return 0;
}
```



Output ?

A. Garbage 10

B. 29 12

C. 29 13

D. 29 14

Practice

```
#include<iostream>
using namespace std;
int fun(int p, int *q, int **r){
    *q++;
    **r*=2;
    p%=11;
    return (p+*q+**r);
}
int main(){
    int *p, a=5;
    p = &a;
    cout<<fun(a, &a, &p)<<" ";
    cout<<a;
    return 0;
}
```

Output ?

A. Garbage 10

B. 29 12

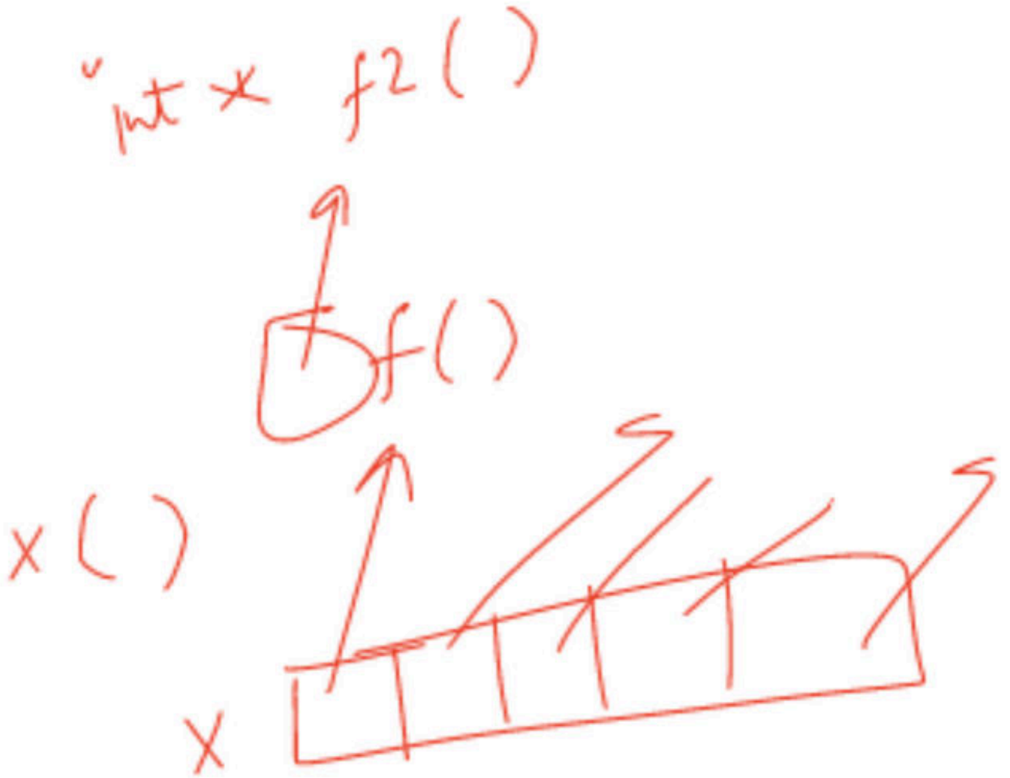
C. 29 13

D. 29 14

Practice

What do the following declarations mean?

1. void (*x[3]) (int)
2. double (*(*x())[]) ()
3. int *(*(*x[5])()) ()



x is a function with no arguments and returns a pointer to an array of pointers to functions with no agument and returns double

Practice

Codechef : https://www.codechef.com/problems/CATSDOGS

Have you tried solving ~~solved~~ this problem?

Chef is a farmer and a pet lover. He has a lot of his favorite pets cats and dogs in the barn. He knows that there are **C** cats and **D** dogs in the barn. Also, one day went to field and found that there were **L** legs of the animals touching the ground. Chef knows that cats love to ride on the dogs. So, they might ride on the dogs, and their legs won't touch the ground and Chef would miss counting their legs. Chef's dogs are strong enough to ride at max two cats on their back.

It was a cold foggy morning, when Chef did this counting. So he is now wondering whether he counted the legs properly or not. Specifically, he is wondering whether it is possible that he counted correctly. Please help Chef in finding it.

Input

First line of the input contains an integer **T** denoting number of test cases. **T** test cases follow.

The only line of each test case contains three space separated integers **C, D, L** denoting number of the cats, number of the dogs and number of legs of animals counted by Chef, respectively.

Output

For each test case, output a single line containing a string "yes" or "no" (both without quotes) according to the situation.

Constraints

- $1 \le T \le 10^5$
- $0 \le C, D, L \le 10^9$

Subtasks

Subtask #1 (20 points)

- $1 \le T \le 10^4$
- $0 \le C, D \le 10^2$

Subtask #2 (30 points)

- $1 \le T \le 10^5$
- $0 \le C, D \le 10^3$

Subtask #3 (50 points)

- Original constraints

```
**Input:**
3
1 1 8
1 1 4
1 1 2
```

```
**Output:**
yes
yes
no
```

Time/Memory CP

Max # legs

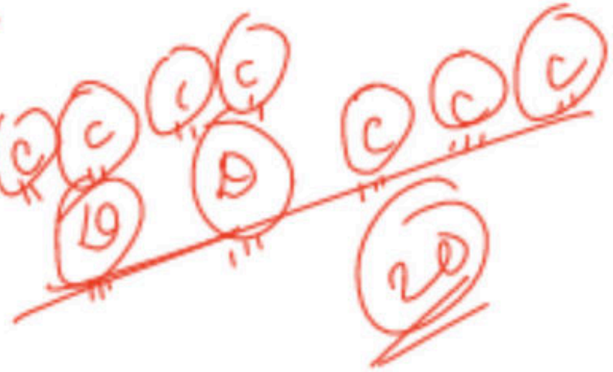
7 cats
2 dogs

Min # legs?

20 ≤ L ≤ 3L

L % 4 == 0

Min # legs?



cats dogs
C, D, L

yes
no

Hint #1

L = 7

1st condition

L % 4 == 0

Hint #2

#cats > 2 # dogs

min # legs # cat

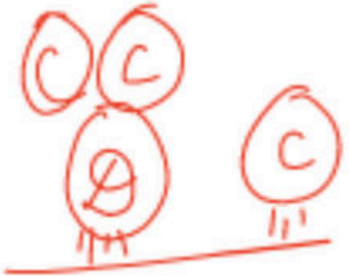
C cats
D dogs

Dog

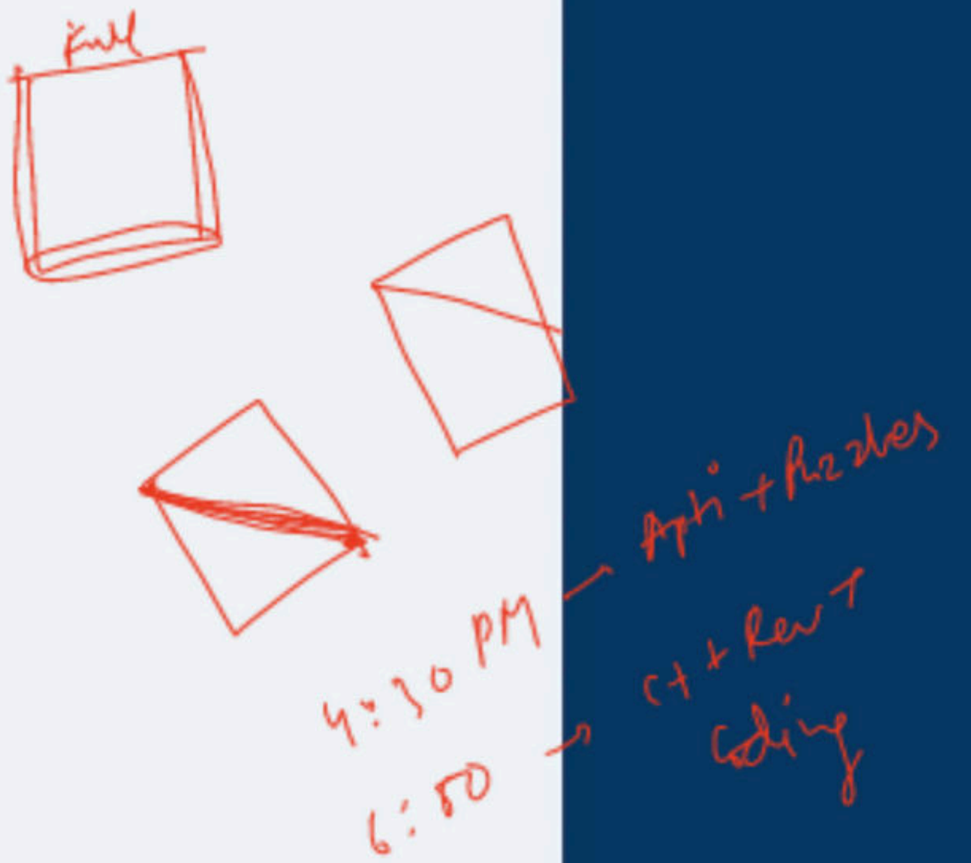
L = 4

C = 3
D = 1

L → 8



unacademy | CODECHEF



TOPIC WISE

BATCH STRUCTURE

TRIUMPH: Beginner to Expert Level Coder in 12 Months (Launching on 8th March)

TOPIC WISE BATCH STRUCTURE

TRIUMPH: Beginner to Expert Level Coder in 12 Months

Week 1

Introduction to Programming

Week 2 - 9

Flow Charts, Loops,Variables & I/O and Online Judges

Week 10 - 13

Sorting & Searching - Concepts And Problem Solving

Week 14 - 19

Basic Data Structures - Fundamentals and Interview Specific Problem Solving

Week 20 - 24

Additional Concepts in C++/Java/Python

Week 25 - 26

Greedy Algorithms with Classical Problem Solving

TOPIC WISE BATCH STRUCTURE

TRIUMPH: Beginner to Expert Level Coder in 12 Months

Week 27-29

Advanced Data Structures - Square Root Decomposition & Complex Problems

Week 30 - 31

Number Theory And Interview Questions

Week 32 - 34

Recursion on DP- Concept to Detail

Week 35 - 39

Discrete Mathematics

Week 40 - 44

Graph Algorithm- Fundamentals to Extensive Problem Solving

Week 45 - 47

Segment Trees

TOPIC WISE BATCH STRUCTURE

TRIUMPH: Beginner to Expert Level Coder in 12 Months

Week 48 - 50

Advanced Dynamic Programming

Week 51 - 53

Computational Geometry

Meet Our Educators

Tanuj Khattar

ACM ICPC World Finalist - 2017, 2018. Indian IOI Team Trainer 2016-2018. Worked @ Google, Facebook, HFT. Quantum Computing Enthusiast.

Sanket Singh

Software Development Engineer @ LinkedIn | Former SDE @ Interviewbit | Google Summer of Code 2019 @ Harvard University | Former Intern @ISRO

Pulkit Chhabra

Codeforces: 2246 | Codechef: 2416 | Former SDE Intern @CodeNation | Former Intern @HackerRank

Riya Bansal

Software Engineer at Flipkart | Former SDE and Instructor @ InterviewBit | Google Women TechMakers Scholar 2018

Triveni Mahatha

Qualified ICPC 2016 World Final. Won multiple Codechef Long Challenges (India). ICPC Onsite Regionals' Problem setter and Judge. IIT Kanpur.

Arnab Chakraborty

Six Sigma Master Black Belt-IQF(USA),AMT by AIMA,NLP & PMP Trained,ITIL-APMG(UK),Control & Automation-ISA(USA),"Star Python"-Star Cert.(USA)

Meet Our Educators

Himanshu Singh

World Finalist ICPC 2020, Winner Techgig Code Gladiators 2020, Winner TCC '19, 2020 CSE Graduate from IIT BHU, Works at Nutanix

Nishchay Manwani

Hey I am Nishchay Manwani from CSE, IIT Guwahati and I'm a Seven star on Codechef and International Grandmaster on Codeforces.

Utkarsh Gupta

I'm a competitive programmer. I am red on CF and 6* on Codechef. I have been teaching CP on various platforms for approx a year.

Arjun Arul

ICPC World Finalist 2012 & 2013, Coach @ IOITC

Deepak Gour

ICPC World Finalist 2020 | Former Instructor @InterviewBit | Software Engineer at AppDynamics

Vivek Chauhan

Codechef: 7 stars (2612) India Rank 6, Codeforces: MASTER (2279), Won Codechef Long Challenges(India), TCO20 Southern Asia Runner up

Link to Unacademy CP course:

https://unacademy.com/goal/competitive-programming/LEARNCP

Link to Triumph batch:

https://unacademy.com/batch/wizard-beginner-to-expert-level-coder-in-12-months/97N4LX9J

JAY BANSAL SIR

Expert in Competitive Programming

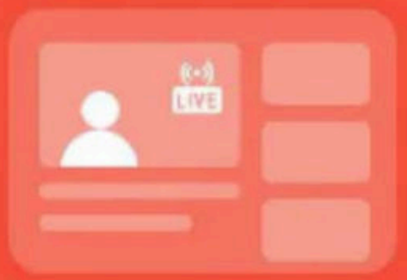
GATE CSE 2019 - AIR 2

Incoming SDE @ Google
ACM ICPC 2019 - AIR 39
B.Tech Gold Medalist

Subjects Taken:

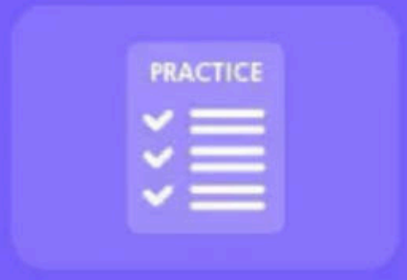
Competitive Programming
C++
Data Structures
Algorithms
Interview Preparation

What You Will Get?



Live Interactive Classes

Attend live interactive classes with our top educators. Interact during class with educators to get all your doubts resolved.

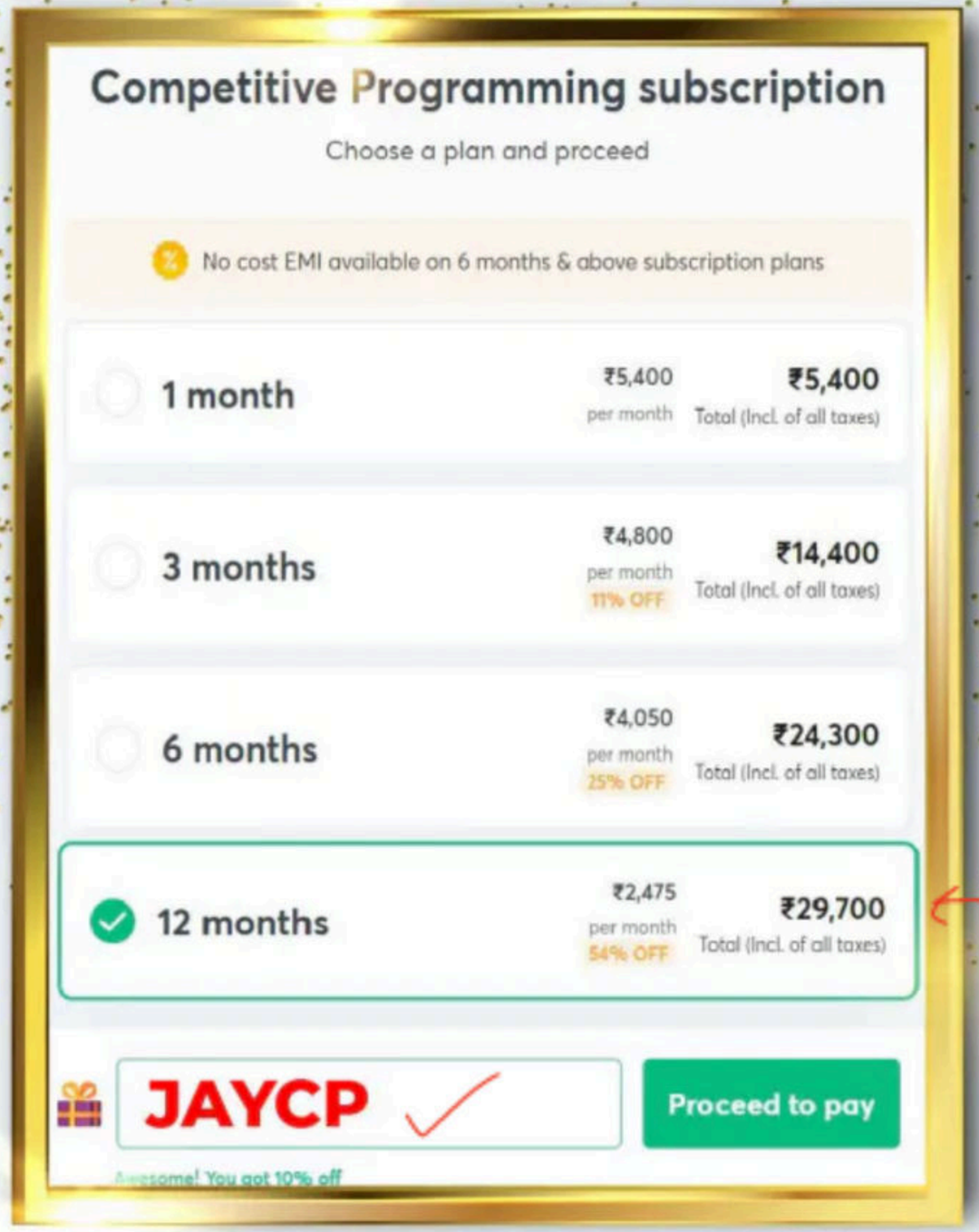


Practice Relevant Problems @ Codechef

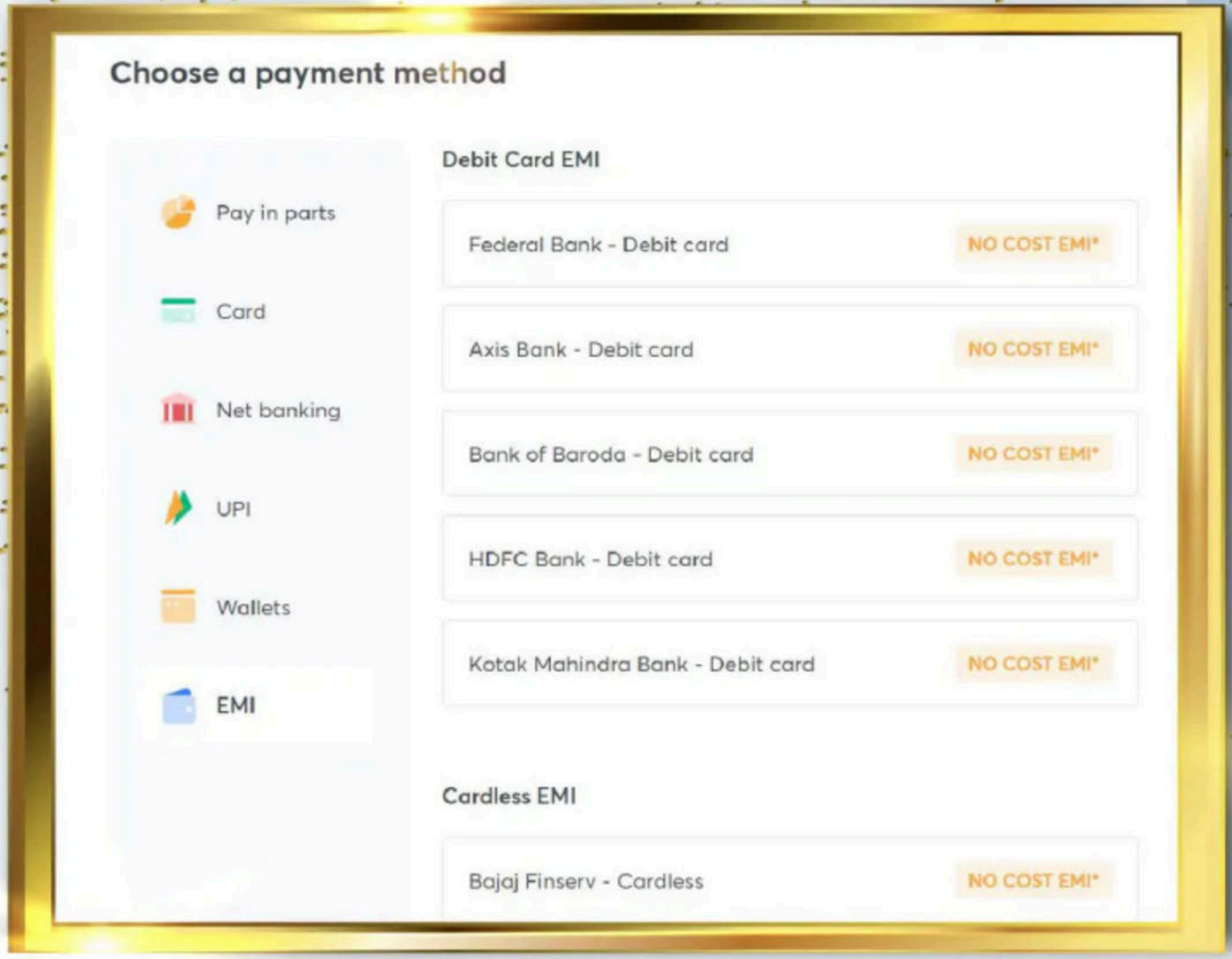
Each class comes with a set of curated practice problems to help you apply the concepts in real time.

Doubt Support

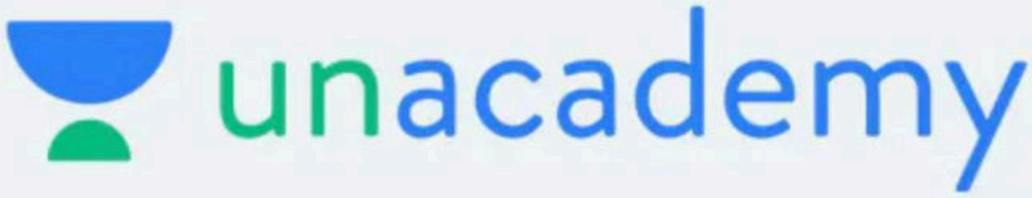
If you get stuck in any problem post class- Get your doubts resolved by our expert panel of teaching assistants and community members instantly.


Competitive Programming subscription
Choose a plan and proceed
No cost EMI available on 6 months & above subscription plans
1 month
₹5,400
per month
₹5,400
Total (Incl. of all taxes)
3 months
₹4,800
per month
11% OFF
₹14,400
Total (Incl. of all taxes)
6 months
₹4,050
per month
25% OFF
₹24,300
Total (Incl. of all taxes)
12 months
₹2,475
per month
54% OFF
₹29,700
Total (Incl. of all taxes)
JAYCP
Proceed to pay
Awesome! You got 10% off

33,880


Choose a payment method
Pay in parts
Card
Net banking
UPI
Wallets
EMI
Debit Card EMI
Federal Bank - Debit card
NO COST EMI*
Axis Bank - Debit card
NO COST EMI*
Bank of Baroda - Debit card
NO COST EMI*
HDFC Bank - Debit card
NO COST EMI*
Kotak Mahindra Bank - Debit card
NO COST EMI*
Cardless EMI
Bajaj Finserv - Cardless
NO COST EMI*

Follow Me on



https://unacademy.com/@jay.bansal

Thank You