

به نام خدا

چگونه خود را برای مسابقات ACM آماده کنیم !!؟

به طور خلاصه ICPC (International Collegiate Programming Contest) به معنی مسابقات بین المللی برنامه نویسی برای دانشجویان دانشگاه ها است .

قوانین این مسابقات به شرح زیر هستند :

- هر تیم از ۳ دانشجو تشکیل می شود
- به هر تیم فقط یک کامپیوتر تعلق می گیرد
- به هر تیم ۵ ساعت وقت داده می شود که ۸ تا ۱۰ سوال را حل نمایند (با زبان های Java , C++ , C و یا احتمالا pascal)
- هر تیمی که بیشترین تعداد سوال را در کمترین زمان حل نماید برنده خواهد بود
- این مسابقات دو مرحله ای هستند : منطقه ای و جهانی . برنده های هر منطقه به مسابقه ی جهانی راه پیدا می کنند

آماده سازی برای مسابقات :

قدم ۰.۱ - مطالعه , مطالعه , مطالعه : روی برنامه نویسی , ساختمان داده ها , الگوریتم ها و برنامه نویسی شی گرا.

قدم ۰.۲ - زبان خود را انتخاب کنید : یک زبان برنامه نویسی انتخاب کنید که با آن راحت هستید. برای مثال می توانید یا C++ یا Java و یا هر دو را انتخاب کنید . (پیشنهاد می کنم که از انتخاب زبان های C و Pascal صرف نظر کنید زیرا این زبان ها فاقد کتابخانه های پیشرفته و قوی هستند)

قدم ۰.۳ - منابع برنامه نویسی جمع آوری کنید : به دنبال کتاب ها و مقالات در مورد برنامه نویسی باشید . منابع و مراجع آنلاین در این زمینه غنی هستند ، از آن ها استفاده کنید.

قدم ۰.۴ - یک محیط برنامه نویسی برای خودتان برپا کنید : اگر قادر به تهیه ی یک لپ تاپ هستید حتما این کار را انجام دهید . در این صورت شما می توانید در هر جایی برنامه نویسی کنید.

بسته به میزبان این مسابقات ، ممکن است مسابقه بر روی سیستم عامل لینوکس (سیستم عاملی دوست داشتنی برای برنامه نویسی) یا ویندوز و یا هر سیستم عامل دیگری برگزار شود .

برای برنامه نویسان Java :

از JDK 1.5 به بالا استفاده کنید زیرا عملیات IO در آن بسیار آسان و ساده شده است. IDE ای که برای این زبان انتخاب می شود قطعاً Eclipse (یک IDE متن باز که توسط IBM پشتیبانی می شود) است که هم بر روی لینوکس و هم ویندوز اجرا می شود. سعی کنید که روش دیباگ (خطایابی) کردن را در این IDE کاملاً یاد بگیرید.

برای برنامه نویسان C/C++ :

انتخاب یک IDE مناسب برای این دو زبان سخت تر است زیرا دامنه ی انتخاب وسیع است :

- در ویندوز می توانید بر روی Visual C++ 2005 تمرین کنید (می توانید این IDE را به طور رایگان از سایت ماکروسافت تهیه کنید)
- هم در ویندوز و هم در لینوکس می توانید از IDE متن باز Eclipse C/C++ Development Tool (CDT) که از کامپایلر GCC/G++ مطعلق به Cygwin استفاده می کند استفاده کنید.

نکته ی مهم برای تمام برنامه نویسان :

- شما باید با طرز استفاده از دیباگر ها آشنایی کامل داشته باشید زیرا که این مساله قدرت برنامه نویسی و تجزیه و تحلیل کد رو در شما افزایش می دهد .
- شما باید با کتابخانه های اساسی زبان خودتون آشنایی کامل داشته باشید . مثلا java API برای برنامه نویسان جاوا و C++ STL برای برنامه نویسان C++ .

قدم ۰.۵ - سایت های داوری و تمرین آنلاین : سایت های تمرین و داوری (online judge) بسیاری وجود دارند که در آرشیو آن ها می توانید صد ها (و حتی هزار ها) سوال از مسابقات پیشین پیدا کنید. شما می توانید در هر زمانی که مایل هستید اقدام به حل کردن سوال کنید و پاسخ های خودتان را در این سایت ها بارگزاری کنید . برنامه شما به صورت خودکار کامپایل و اجرا می شود و به دقت مورد تجزیه و تحلیل قرار می گیرد . وضعیت اجرا از قبیل : " قبول شد (accepted) " ، " پاسخ اشتباه (wrong answer) " ، " خطای کامپایل (compile error) " ، " خطای شیوه ی ارایه (presentation error) " ، " عبور از حد مجاز زمان (time limit exceeded) " ، " عبور از حد مجاز حافظه (memory limited exceed) " ، " عبور از حد مجاز خروجی (output limit exceed) " به شما نشان داده خواهد شد . حتی در بعضی از سایت ها اگر خطای کامپایل در برنامه وجود داشته باشد پیغام خطای کامپایل را به نشان می دهد .

در زیر تعدادی از سایت های معروف در این زمینه را معرفی کرده ام (برای بدست آورد فهرست کاملی از این سایت ها می توانید عبارات " ICPC " و یا " online judge " را در گوگل جستجو کنید) :

[Peking University Online Judge \(PKU\)](#) : این سایت از زبان های زیادی پشتیبانی می کند از جمله Java (JDK 1.5), GNU's GCC/G++ (for C/C++) and Visual C/C++ version 6

[Universidad de Valladolid Online Judge \(UVA\)](#) : قابل اطمینان ترین سایت با یک فروم خوب (که به یک موتور جستجو مجهز است) . پشتیبانی این سایت از زبان C++ بسیار خوب است. اگرچه در حد متوسط از زبان جاوا پشتیبانی می کند (نمی توانید از JDK 1.5 استفاده کنید) .

[USA Computing Olympiad \(USACO\) Training Program](#) : این یک سایت تمرینی - آموزشی برای IOI (International Olympiad in Informatics for high school students) است. این سایت دوره های آموزشی اصولی و قاعده داری را در زمینه های الگوریتم های پر استفاده از قبیل : کوتاهترین مسیر ، گریدی ، برنامه نویسی پویا و ... ارایه می دهد . این سایت از زبان های C++ و Java JDK 1.5 پشتیبانی می کند.

به صورت واقعی شروع کنیم :

قدم اول - PKU Online Judge را امتحان کنید

۱. در این سایت به آدرس <http://acm.pku.edu.cn/JudgeOnline/> PKU online judge ثبت نام کنید.
۲. قسمت FAQ را برای پی بردن به قوانین ارسال پاسخ ها مطالعه کنید
۳. دوباره قسمت FAQ را مطالعه کنید.
۴. قواعد برنامه نویسی در ICPC به صورت زیر هستند :

برنامه نویسان جاوا :

۱. ورودی ها از System.in می آیند و خروجی ها به System.out می روند (File IO مجاز نیست).

۲. فایل سورس باید حاوی یک کلاس به نام Main باشد که در این کلاس یک متد با نام main که دارای آرگومان های ورودی است به صورت زیر قرار دارد :

```
main:
public static void main(String[] args) { ... }.
```

۱. برنامه نویسان C++ :

۲. ورودی ها از std:cin می آیند و خروجی ها به std:cout می روند (File IO مجاز نیست).

۳. فایل سورس باید حاوی یک تابع با نام main به صورت زیر باشد :

```
int main() { ... }.
```

۵. مساله ی (A+B) 1000 که جواب آن در قسمت FAQ قرار داده شده است را حل کنید. هدف این مساله این است که شما قوانین برنامه نویسی ای که در بالا ذکر شد را درک کنید. هنگام فرستادن جواب دقت کنید که زبان برنامه نویسی خود را درست انتخاب کنید. برنامه نویسان جاوا باید از JDK 1.5 یا بالاتر استفاده کنند. از Scanner ، in.nextInt() ، In.nextDouble() و in.next() برای خواندن int ، double و String و از System.out.printf ("formatString",args ..) برای خروجی استفاده کنید.

یک کد نمونه ی JDK 1.5 برای ICPC :

```
import java.util.Scanner;

public class Main { // save as Main.java
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        int i = in.nextInt(); // read int
        double d = in.nextDouble(); // read double
        String s = in.next(); // read String
        // There is no nextChar(), use next() and charAt()
        char c = s.charAt(2);
        // Read whole line (or rest of the line past '\n')
        String line = in.nextLine();

        System.out.printf("%4d, %6.2f, %s, %c\n", i, d, s, c);
        // Use %f for double (not %lf)
        // Don't forget to print the '\n'
    }
}
```

۶. مثال های ساده دیگر را هم حل کنید.(برای اینکه مثال های ساده را پیدا کنید می توانید به درصد پذیرفته شده ها نگاه کنید .)

قبل از اینکه مثال های دیگری را حل کنید دوباره قواعد برنامه نویسی را بخوانید.

ورودی ها و خروجی ها برنامه ها باید دقیقاً مطابق چیز هایی که گفته شد باشند. نباید در خروجی چیزی جر جواب مساله چاپ شود . مثلاً نمی توان در خروجی عبارتی شبیه به "Please enter a number" را چاپ کنید زیرا کسی پیغام شما را نمی خواند و همه پیز توسط کامپوتر انجام می شود.

۷. مثال "1004 (Financial Management)" را حل کنید . این مثال در رابطه با پیدا کردن میانگین ۱۲ عدد است.

نکته : برای آزمایش این برنامه دو راه دارید : یک اینکه خود ۱۲ عدد را به برنامه بدهید و خروجی را چک کنید (که راه کندی هست) و دو اینکه ۱۲ عدد را در یک فایل مثلا با نام "in.txt" ذخیره کنید ، بعد cmd را اجرا کنید و سپس با استفاده از عملگر پایپ که به شکل "<" است فایل in.txt را به ورودی برنامه بفرستید.

برای برنامه نویسان جاوا :

فرض کنید نام سورس برنامه Main.java هست

```
> javac Main.java  
> java Main < in.txt
```

برای برنامه نویسان C++ :

فرض کنید که نام سورس برنامه test.cpp است

```
> g++ -o test.exe test.cpp  
> test < in.txt
```

۸. مثال "1003 (Hangover)" را حل کنید . این مثال در رابطه با حساب کردن یک سری هارمونیک است.
۹. مثال "1005 (I think I need a house boat)" را حل کنید. این مثال در رابطه با حساب کردن محیط یک نیم دایره است.

قدم دوم - UVA Online Judge را امتحان کنید

۱. این سایت فقط برای برنامه نویسان C/C++ است . برنامه نویسان جاوا باید می توانند این سایت را فراموش کنند.
۲. در این سایت ثبت نام کنید : [UVA online judge @ http://online-judge.uva.es/problemset/](http://online-judge.uva.es/problemset/)
۳. قسمت HOWTOs را مطالعه کنید. خصوصا قسمتی که در مورد چگونگی فرستادن جواب ها توضیح داده است.
۴. مساله ی "100 (3n+1)" را حل کنید و با جوابی که در قسمت HOWTOs گذاشته شده مقایسه کنید.

قدم سوم - USACO Training Problem را امتحان کنید

۱. در این سایت ثبت نام کنید : [USACO Training Program @ http://train.usaco.org/usacogate/](http://train.usaco.org/usacogate/)
۲. قسمت های "Section 1.0 Text Introduction" و "Section 1.1 Submitting Solutions" را مطالعه کنید. همانطور که در این قسمت ها گفته شده این سایت از IOI پشتیبانی می کند . پس روش ارسال پاسخ ها در آن با سایت های بالا که از ICPC پشتیبانی می کردند متفاوت است.
۳. شما باید از یک اطلاعات ورودی را از یک فایل که "xxxx.in" نام دارد بخوانید و اطلاعات خروجی را توی فایلی با نام "xxxx.out" بنویسید که در اینجا xxxx همان نام مساله است.

برای برنامه نویسان جاوا :

راه حل نمونه ای که سایت برای برنامه های جاوا در اختیاران قرار داده مبتنی بر JDK 1.2 است که در عمل File IO دارای سختی های خاص خودش است . همان مثال در JDK 1.5 به صورت زیر است :

Program Template for USACO

```
/*  
ID: yourID  
LANG: JAVA
```

```

TASK: test
*/

import java.util.Scanner;
import java.util.Formatter;
import java.io.File;
import java.io.IOException;

public class test { // saved as test.java
    public static void main (String [] args) throws IOException {
        Scanner in = new Scanner(new File("test.in")); // file input
        Formatter out = new Formatter(new File("test.out")); // file output

        int a = in.nextInt();
        int b = in.nextInt();
        out.format("%d\n", a+b); // format() has the same syntax as printf()

        out.close(); // flush the output and close the output file
        System.exit(0); // likely needed in USACO
    }
}

```

به همین روند ادامه بدهید و سعی کنید که مساله های موجود در قسمت training problem را حل کنید. همانطور که گفته شد، این سایت آموزش های اصولی ای را در رابطه با الگوریتم هایی که مکررا در مسابقات برنامه نویسی استفاده می شوند دارد.

برای شرکت در مسابقات برنامه نویسی دانستن موارد زیر بسیار مفید و کار آمد است :

- داشتن یک سطح دانش پایه ای در مورد ساختمان داده ها (مثل Vector, Linked list , queue , stack)
- الگوریتم های زیادی را باید بدانید (می توانید در سایت USACO چند الگوریتم پایه ای را که حتما باید بدانید ببینید) :
- همه ی الگوریتم های مرتب سازی
- روش استفاده از bit-operations (این لینک را نگاه کنید "[tips, tricks and tweak](#)")
- تحلیل الگوریتم های معروف
- گراف (BFS ، DFS ،)
- ریاضی (نظریه ی اعداد)
- هندسه (Convex Hull ، Interval tree ،)
- الگوریتم گریدی
- الگوریتم دینامیک
- و دیگر چیز ها

مبتدی ها :

- مساله های "ad-hoc" و مساله های ساده ای که مربوط به ریاضیات هستند را حل کنید (برای مثال gcd ، فیبوناچی ، اعداد اول) روی تولید اعداد اول به سریعترین روش کار کنید ("[tips, tricks and tweaks](#)"). روی کار با رشته ها و عملیات ها روی اعداد بسیار بزرگ کار کنید(البته بدون استفاده از کلاس یا کتابخانه ی BigInteger)
- از STL C++ یا API JAVA در این مرحله استفاده نکنید.

پیشرفته ها :

- باید STL C++ یا API Java را به صورت کامل یاد بگیرید .

- گراف ها (bfs, dfs, flood fill algorithm, shortest path (diskja, floyd), tree, network flow)
- Dynamic algorithm, dictionary algorithm
- هندسه (حداقل باید convex-hull را بلد باشید ، فهمیدن اینکه آیا یک نقطه نزدیک یک چند گوش است یا نه ، روش حساب کردن محیط یک چند گوش و ...)
- Graphs, dynamic, ad-hoc, simulation, maths, geometry + dynamics, graphs + geometry, maths + dynamic, and so on

موفق باشید

منبع : www.acmsolver.org

مترجم : علیرضا داودی

کپی برداری با ذکر منبع و نام مترجم مجاز است