

# مسئله‌ی برنامه‌نویسی چهارشنبه

می‌فهمیم بهترین باشیم...

هیچ کس، حتی خوش‌بین‌ترین عضو تیم فنی این رقابت هم تصور نمی‌کرد که پنج‌شنبه‌های سخت در مدت پنج ماه تا این حد در جهان مطرح شود و علاقمندان آن هر هفته را با انتظار برای فرا رسیدن پنج‌شنبه سپری کنند. اما با توجه به هدف‌های فصل یکم پنج‌شنبه‌های سخت، در ماه‌های اخیر جای خالی رقابت دیگری حس می‌شد تا در نهایت تیم فنی تصمیم گرفت که رقابت جدیدی را برای پاسخ به این نیاز در کنار پنج‌شنبه‌های سخت پایه‌گذاری کند. بنابراین، همزمان با شروع نیمسال جدید، مفتخر هستیم تا رقابت جدیدی را با نام «مسئله‌ی برنامه‌نویسی چهارشنبه» معرفی کنیم؛ رقابتی که، با استناد بر پیش‌بینی‌ها، بسیار پرشکوه برگزار می‌شود. یکی از تفاوت‌های این رقابت نسبت به پنج‌شنبه‌های سخت این است که هر کسی می‌تواند مسئله‌های این رقابت را طرح کند.

طرح سؤال در چهارشنبه‌ها سه گام اصلی دارد: گام اول فرستادن متن سؤال به سردبیر چهارشنبه‌ها ([gholamirudi@nit.ac.ir](mailto:gholamirudi@nit.ac.ir))، گام دوم نوشتن برنامه‌ای که نمونه‌های ورودی را تولید کند و گام سوم نوشتن برنامه‌ای که به خروجی مسئله امتیاز دهد. در ادامه‌ی این مستند، این سه گام برای یک مسئله‌ی نمونه نشان داده می‌شوند.

هر روز سردبیر پنجشنبه‌های سخت، از دو صندوق پست الکترونیکی نامه دریافت می‌کند. با گرفتن تعداد نامه‌های هر یک از این دو صندوق، تعداد کل نامه‌ها را چاپ کنید. فرض کنید حداکثر یک میلیون نامه در هر صندوق دریافت می‌شود و ممکن است تعداد نامه‌ها منفی باشد! یک نمونه‌ی ورودی و خروجی این مسئله در جدول زیر نمایش داده شده است.

ورودی	خروجی
۱ ۲	۳

برنامه‌ی تولید نمونه‌ها یک عدد را به عنوان ورودی دریافت می‌کند و نمونه‌ی ورودی مربوط به آن را تولید می‌کند. این برنامه می‌تواند به زبان C، C++ یا Python نوشته شود.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    int n;
    if (argc < 2)
        return 1;
    n = atoi(argv[1]);
    if (n == 0) /* case 0: testing zeros */
        printf("0 0\n");
    if (n == 1) /* case 1: normal operands */
        printf("1 2\n");
    if (n == 2) /* case 2: negative numbers */
        printf("-1 2\n");
    if (n == 3) /* case 3: negative results */
        printf("1 -2\n");
    if (n == 4) { /* case 4: random input */
        srand(4);
        printf("%d %d\n", rand() % 1000 - 500,
                rand() % 1000 - 500);
    }
    if (n > 4)
        return 1;
    return 0;
}
```

برنامه‌ی داور، آدرس فایل ورودی (پارامتر اول) و آدرس فایل خروجی یک برنامه (پارامتر دوم) را دریافت می‌کند و امتیاز برنامه را به صورت یک عدد در خروجی چاپ می‌کند. برنامه‌ی زیر این کار را برای مسئله‌ی نمونه‌ی این مستند انجام می‌دهد. در صورتی که جمع درست محاسبه شده باشد، امتیاز یک چاپ می‌شود و در غیر این صورت، با برگشت مقدار یک از تابع `main` خطا گزارش می‌شود.

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    FILE *ifp, *ofp;
    int a, b, c;
    if (argc < 3)
        return 1;
    ifp = fopen(argv[1], "r");
    ofp = fopen(argv[2], "r");
    if (!ifp || !ofp)
        return 1;
    if (fscanf(ifp, "%d %d", &a, &b) != 2)
        return 1;      /* bad input */
    if (fscanf(ofp, "%d", &c) != 1)
        return 1;      /* bad output */
    if (a + b != c)
        return 1;      /* wrong answer */
    printf("%d\n", 1); /* print the score */
    return 0;
}
```