

কম্পিউটার প্রোগ্রামিং বই

বাংলা ভাষায় কম্পিউটার প্রোগ্রামিং শেখার বই। বইটিতে প্রোগ্রামিং ভাষা হিসেবে সি (C) ব্যবহার করা হয়েছে। প্রোগ্রামিংয়ের জগতে যারা নতুন, তাদের জন্য বইটি সহায়ক। ওয়েবসাইটের যেকোনো অংশ অবাণিজ্যিক উদ্দেশ্যে কপি, প্রিন্ট ও বিতরণ করা যাবে (সৃজনী সাধারণ অবাণিজ্যিক লাইসেন্স)।

সূচিপত্র	বইয়ের ভূমিকা	ভিডিও লেকচার	প্রোগ্রামিং সমস্যা	প্রোগ্রামিং আর্টিকেল	FAQ	ফেসবুক পেজ	PDF Download
----------	---------------	--------------	--------------------	----------------------	-----	------------	--------------

প্রোগ্রামিং সংক্রান্ত নানান বই ঘরে বসে অনলাইনে অর্ডার করতে [ক্লিক করুন এখানে](#)

বাংলা ভাষায় পাইথন দিয়ে প্রোগ্রামিং শেখার বই - <http://pybook.subeen.com>

[প্রোগ্রামিং বইঃ অধ্যায় ছয়] অ্যাারে।

এতক্ষণে তোমাদের প্রোগ্রামিং জ্ঞান-বুদ্ধি একটু বেড়েছে। চলো, এবার তাহলে কিছু জনসেবামূলক কর্মকাণ্ড করা যাক। আমরা স্কুলের প্রিয় গণিত শিক্ষকের জন্য পরীক্ষার ফলাফল বের করার প্রোগ্রাম লিখে দেব। ওই স্কুলে প্রথম সাময়িক, দ্বিতীয় সাময়িক ও বার্ষিক এই তিনটি পরীক্ষাই 100 নম্বরের হয়। তারপর বার্ষিক পরীক্ষার 50%, দ্বিতীয় সাময়িক পরীক্ষার 25% ও প্রথম সাময়িক পরীক্ষার 25% নিয়ে চূড়ান্ত ফলাফল প্রকাশ করা হয়। তাহলে আমাদের প্রোগ্রামের ইনপুট হচ্ছে ওই তিনটি পরীক্ষার নম্বর। আমাদেরকে চূড়ান্ত ফলাফল দেখাতে হবে। এটি কোনো ব্যাপারই নয়:

```
#include <stdio.h>
int main()
{
    int ft_marks, st_marks, final_marks;
    double total_marks;
    ft_marks = 80;
    st_marks = 74;
    final_marks = 97;
    total_marks = ft_marks / 4.0 + st_marks / 4.0 + final_marks / 2.0;
    printf("%0.01f\n", total_marks);
    return 0;
}
প্রোগ্রাম: ৬.১
```

প্রোগ্রামটির আউটপুট 87। (কিন্তু আমি যদি $total_marks = ft_marks / 4.0 + st_marks / 4.0 + final_marks / 2.0$; না লিখে এভাবে লিখতাম $total_marks = ft_marks / 4 + st_marks / 4 + final_marks / 2$; তাহলে আউটপুট আসে 86। কারণ কী? কম্পিউটারের মাথা খারাপ নাকি আমার?)

আমরা কিন্তু আমাদের প্রিয় শিক্ষকের তেমন কোনো উপকার করতে পারলাম না। কারণ তাঁর ক্লাসে মোট ছাত্রছাত্রীর সংখ্যা চল্লিশ। তাহলে স্যারকে চল্লিশবার প্রোগ্রামটি চালাতে হবে! কিন্তু এটি তো কোনো কাজের কথা হলো না। আমাদের উচিত, সবার চূড়ান্ত ফলাফল একটি প্রোগ্রামের মাধ্যমে নির্ণয় করা। তেমন কোনো কঠিন কাজ নয় এটি। আমরা এমন একটি প্রোগ্রাম লেখা শুরু করে দিতে পারি:

```
#include

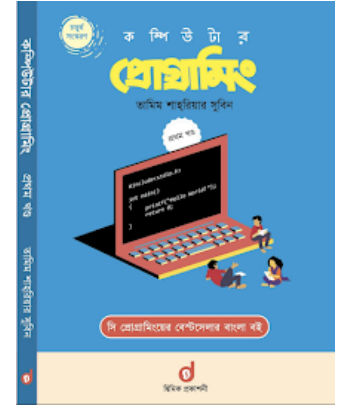
int main()
{
    int ft_marks_1, st_marks_1, final_marks_1, ft_marks_2, st_marks_2, final_marks_2, ft_marks_3,
    st_marks_3, final_marks_3,
```

তোমরা নিশ্চয়ই বুঝতে পারছ, আমি কী করতে যাচ্ছি? বলো তো এভাবে প্রোগ্রামটি লিখতে গেলে মোট কয়টি ভেরিয়েবলের দরকার? 160টি। স্যারের কষ্ট কমাতে গিয়ে আমাদের কষ্ট এত বাড়ানোর কোনো মানে হয় না। কিন্তু এধরনের প্রোগ্রাম তো আমাদের প্রায়ই লিখতে হবে। চিন্তা নেই! প্রায় সব প্রোগ্রামিং ল্যাংগুয়েজেই অ্যারে (Array) নামে একটি চমৎকার জিনিস আছে। এতে একই ধরনের অনেকগুলো ভেরিয়েবল একসঙ্গে রাখা যায়। ভেরিয়েবলের যেমন নাম রাখি, অ্যারের বেলাতেও তেমন একটি নাম দিতে হয়। Cতেও অ্যারে আছে।

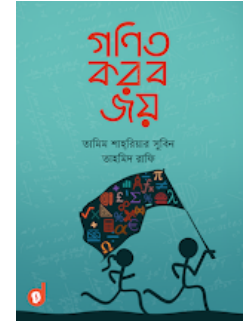
ভেরিয়েবলের যেমন একটি ডাটা টাইপ থাকে, অ্যারেরও থাকে। অ্যারেটি যে ডাটা টাইপের হবে তাতে কেবল সেই রকম ডাটাই রাখা যাবে। যেমন char টাইপের অ্যারেতে কেবল char টাইপের জিনিস থাকবে।

অ্যারেতে কয়টি উপাদান থাকবে সেটি শুরুতেই বলে দিতে হয়।

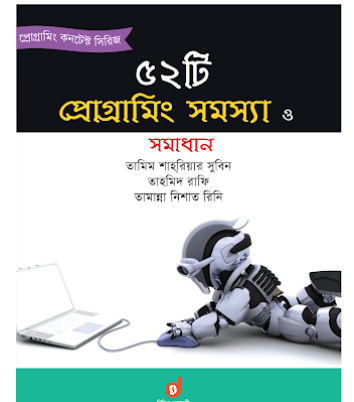
কম্পিউটার প্রোগ্রামিং ১ম খণ্ড বইয়ের হার্ডকপি



বই সম্পর্কে বিস্তারিত জানতে নিচের ছবিতে ক্লিক করুন



৫২টি প্রোগ্রামিং সমস্যা ও সমাধান



বইটি সম্পর্কে আরো জানতে ওপরের ছবিটিতে ক্লিক করুন।

int ara[10]; এভাবে আমরা একটি অ্যারে ডিক্লেয়ার করতে পারি, যার নাম হচ্ছে ara, যেটিতে কেবল ইন্টিজার টাইপের ডাটা থাকবে আর এই অ্যারেতে মোট দশটি সংখ্যা রাখা যাবে। প্রথমটি হচ্ছে ara[0] (হ্যাঁ, ara[1] না কিন্তু), দ্বিতীয়টি ara[1], তৃতীয়টি ara[2], এভাবে দশম সংখ্যাটি হচ্ছে ara[9]। অর্থাৎ, ara[i] হচ্ছে i+1তম উপাদান।

এবারে চলো অ্যারে নিয়ে একটু খেলাধুলা করা যাক। প্রতিটি প্রোগ্রাম কিন্তু অবশ্যই কম্পিউটারে চালিয়ে দেখবে।



```
#include <stdio.h>
int main()
{
    int ara[5] = {10, 20, 30, 40, 50};
    printf("First element: %d\n", ara[0]);
    printf("Third element: %d\n", ara[2]);
    return 0;
}
```

প্রোগ্রাম: ৬.২

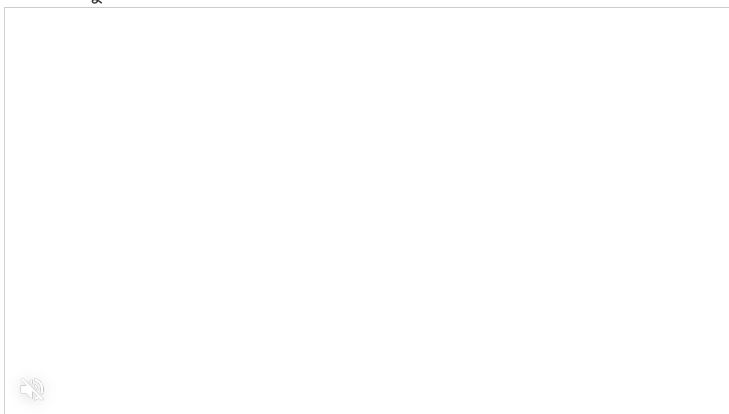
আউটপুট ঠিকঠাক দেখতে পাচ্ছ?

আরেকটি প্রোগ্রাম:

```
#include <stdio.h>
int main()
{
    int ara[5] = {6, 7, 4, 6, 9};
    printf("%d\n", ara[-1]);
    printf("%d\n", ara[5]);
    printf("%d\n", ara[100]);
    return 0;
}
```

প্রোগ্রাম: ৬.৩

এটির জন্য কী আউটপুট আসা উচিত? আমি জানি না এবং এটি জানা সম্ভব নয়। যেকোনো ধরনের সংখ্যা আসতে পারে। এগুলোকে গারবেজ (garbage) বলে। কারণ আসলে তো ওই অ্যারেতে -1, 5, 100 এই ইনডেক্স বলতে কিছু নেই। অ্যারেটির দৈর্ঘ্যই হচ্ছে 5 সুতরাং ইনডেক্স হবে 0 থেকে 4।



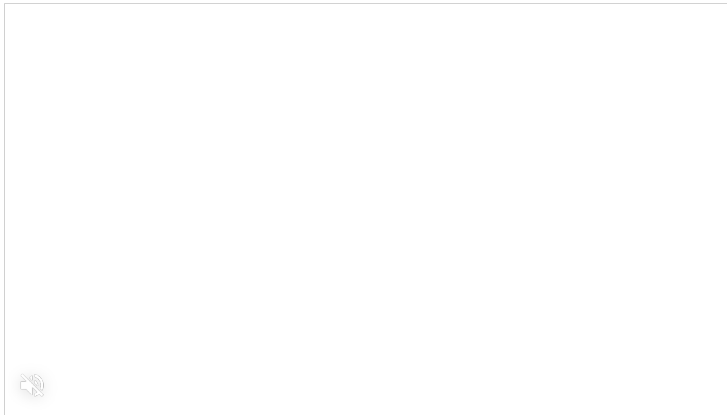
প্রোগ্রামিং প্রোগ্রামিং

সমস্যা string সি number theory অ্যারে কম্পিউটার প্রোগ্রামিং কম্পাইলার ধারার যোগফল প্রাইম নাম্বার মৌলিক সংখ্যা prime number অক্ষর আউটপুট ইনপুট কোডব্লকস্ জাভা জোড় জ্যামিতি পাইথন পারফেক্ট নাম্বার প্রোগ্রামিং প্রতিযোগিতা প্রোগ্রামিং বই পয়েন্টার ফাংশন বিজোড় মেমোরি যোগ্য সংখ্যা সংখ্যা সাবস্ট্রিং সি প্লাস প্লাস 2-D array Armstrong C C++ Multiples X এর গুণিতক alphabet array beginning computer programming computer programming descending desktop diamond shape divisor freopen function geometry laptop loop lowercase multidimensional array palindrome prime programming problems reverse sorting sub-string uppercase অঙ্কগাণিত্য অক্ষরের ঘনঘটা অধোগামী অ্যান্ড্রয়েড আইডিই আইসিটি আউটসোর্সিং আর্মস্ট্রং সংখ্যা ইনফরমেটিক্স অনিস্পিয়ড ইনস্টল উল্টে দেখা একান্তর উপাদান এলিয়েন এলোমেলো অ্যারে এসিএম আইসিপিসি ওয়েবসাইট কনটেন্ট কভিশনাল লজিক কম্পিউটার কম্পিউটার সায়োল কোড কাতা ক্যারেক্টার ক্রম গুণনীয়ক গুণিতক গৌণিক চিহ্ন পরিচয় জিসিসি ডাটা টাইপ ডিভিডি ডেস্কটপ ত্রিভুজের ক্ষেত্রফল ধারা ধারার যোগফল-২ ধারার যোগফল-৩ পূর্ণসংখ্যা প্যালিনড্রোম প্যাসকেল প্যাসকেলের ত্রিভুজ প্রবাবিলিটি প্রোগ্রামিং কনটেন্ট প্রোগ্রামিং ক্যারিয়ার প্রোগ্রামিং প্রবলেম প্রোগ্রামিং ল্যাসুয়েজ প্রোগ্রামিং শেখা প্রোবাবিলিটি ফাইল ফাইল আউটপুট ফাইল ইনপুট ফ্যাক্টোরিয়াল বই বর্গ বর্গক্ষেত্র বর্ণমালা থেকে সংখ্যা বাইনারি নাম্বার বাইনারি সংখ্যা বাইনারি সংখ্যা পদ্ধতি বাইনারি সার্চ বাক্স বিভাজনসাধ্য-১ বিভাজনসাধ্য-২ বৃত্ত বৃত্তের বাইরে ভাজক যোগফল যোগফল নির্ণয় যোগ্য রান রেট লসাগু লুপ ল্যাপটপ শব্দ বিপর্যয় শব্দ সাজানো সংখ্যা বিপর্যয় সমাধান সম্ভাব্যতা সচিং সি প্রোগ্রাম সি প্রোগ্রামিং সি ল্যাসুয়েজ সিএসই সিড স্বরবর্ণ গণনা স্বরবর্ণ-ব্যঞ্জনবর্ণ হীরক রাজ্য

এখন কোনো অ্যারের সব উপাদান যদি একসঙ্গে দেখাতে চাই, তাহলে উপায় কী? উপায় হচ্ছে প্রথম উপাদান (ara[0]), দ্বিতীয় উপাদান (ara[1]), তৃতীয় উপাদান (ara[2]) ... এভাবে একে একে সবগুলো প্রিন্ট করা। আর তার জন্য অবশ্যই আমরা লুপের সাহায্য নেব।

```
#include <stdio.h>
int main()
{
    int ara[10] = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};
    int i;
    for(i = 0; i < 10; i++) {
        printf("%d th element is: %d\n", i+1, ara[i]);
    }
    return 0;
}
```

প্রোগ্রাম: ৬.৪





আর যদি শেষ উপাদান থেকে প্রথম উপাদান পর্যন্ত দেখাতে হতো? কোনো সমস্যা নেই, শুধু লুপে এ index টি 9 থেকে 0 পর্যন্ত আনলেই চলবে। এখন তোমরা প্রোগ্রামটি লিখে ফেলো।

এবারে একটি ছোট সমস্যা। কোনো একটি অ্যারেতে দশটি উপাদান আছে, সেগুলো বিপরীত ক্রমে রাখতে হবে। অর্থাৎ দশম উপাদানটি হবে প্রথম উপাদান, প্রথমটি হবে দশম, দ্বিতীয়টি হবে নবম, নবমটি হবে দ্বিতীয়.. এই রকম। তার জন্য আমরা যেটি করতে পারি, আরেকটি অ্যারের সাহায্য নিতে পারি। দ্বিতীয় অ্যারেটিতে প্রথম অ্যারের উপাদানগুলো বিপরীত ক্রমে রাখবো। তারপর দ্বিতীয় অ্যারেটি প্রথম অ্যারেতে কপি করে ফেলব।


```
#include <stdio.h>
int main()
{
    int ara[] = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};
    int ara2[10];
    int i, j;
    for(i = 0, j = 9; i < 10; i++, j--) {
        ara2[j] = ara[i];
    }
    for(i = 0; i < 10; i++) {
        ara[i] = ara2[i];
    }
    for(i = 0; i < 10; i++) {
        printf("%d\n", ara[i]);
    }
    return 0;
}
```


প্রোগ্রাম: ৬.৫





Square Online
for restaurants.

[Learn more](#)









Useful Sites

[সুবিন ডট কম](#)

[ফ্রি পাইথন বই](#)

[পাইথন নিয়ে বাংলা লেকচার](#)

[দ্বিমিক কম্পিউটিং স্কুল](#)

[দ্বিমিক প্রকাশনী](#)

[My Python Blog](#)

[শাফায়েতের প্রোগ্রামিং ব্লগ](#)

মোট পৃষ্ঠাদর্শন

4 4 2 0 8 0
6



এখানে লক্ষ করো যে প্রথম অ্যারেটির ক্ষেত্রে আমি তৃতীয় বন্ধনীর ভেতর অ্যারের উপাদান সংখ্যা বলে দিইনি, কারণ সি-এর কম্পাইলার দ্বিতীয় বন্ধনীর ভেতর সংখ্যাগুলো দেখেই বুঝে নিতে পারে যে `ara`তে দশটি উপাদান আছে। দ্বিতীয় অ্যারে অর্থাৎ `ara2`তে এখন কোনো কিছু নেই। তাই শুরুতেই বলে দিতে হবে যে তাতে কয়টি উপাদান থাকবে। তাহলে কম্পাইলার সেই অনুসারে কম্পিউটারের মেমোরির মধ্যে অ্যারের জন্য জায়গা করে নেবে।

প্রোগ্রামটি ভালোভাবেই কাজ করছে। কিন্তু তোমরা একটু চিন্তাভাবনা করলেই বুঝতে পারবে যে দ্বিতীয় অ্যারেটি ব্যবহার করার কোনো দরকার ছিল না। আমরা একটি বহুল প্রচলিত পদ্ধতিতেই কাজটি করতে পারতাম।

`int temp;`

`temp = ara[9];`

`ara[9] = ara[0];`

`ara[0] = temp;`

প্রথম ও দশম উপাদান অদলবদল হয়ে গেল। তারপর

`temp = ara[8];`

`ara[8] = ara[1];`

`ara[1] = temp;`

দ্বিতীয় ও নবম উপাদান অদলবদল হয়ে গেল। তাহলে চলো প্রোগ্রামটি লিখে ফেলি:

```
#include <stdio.h>
int main()
{
    int ara[] = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100};
    int i, j, temp;
    for(i = 0, j = 9; i < 10; i++, j--) {
        temp = ara[j];
        ara[j] = ara[i];
        ara[i] = temp;
    }
    for(i = 0; i < 10; i++) {
        printf("%d\n", ara[i]);
    }
    return 0;
}
```

প্রোগ্রাম: ৬.৬

প্রোগ্রামটি চালাও। কী দেখলে? আউটপুট কি এরকম?

```
10
20
30
40
50
60
70
80
90
100
```

তারমানে কাজ হয়নি! আসলে আমি একটি ছোট্ট ভুল করেছি, সেটি তোমরা খুঁজে বের করো। এ ধরনের ভুলকে বলে বাগ (bug), তখন প্রোগ্রাম ঠিকমতো রান করে কিন্তু সঠিক আউটপুট দেয় না। আমার কোডে বাগ আছে, তোমরা ডিবাগ (debug) করো (মানে বাগটি বের করে ঠিক করো)।

এখন চলো আমাদের আগের সমস্যায় ফিরে যাই। আমরা এখন প্রথম সাময়িক পরীক্ষায় সবার গণিতের নম্বর একটি

আশা করি, তোমাদের মধ্যে কেউ কেউ প্রোগ্রামটি লিখে ফেলেছ। যদি কমপক্ষে এক ঘণ্টা চেষ্টার পরেও লিখতে না পারো তাহলে এখন আমরা সমাধানের চেষ্টা করতে পারি। শুরুতেই একটি ব্যাপার খেয়াল করো যে কেউ কিন্তু 50-এর নিচে নম্বর পায়নি। তাই 50 থেকে 100 পর্যন্ত কোন নম্বর কতজন পেল সেটি বের করলেই চলবে। আমার মাথায় প্রথমেই যে সমাধান আসছে সেটি হলো total_marks অ্যারেতে প্রথমে দেখব, কয়টি 50 আছে, তারপর আবার দেখব কয়টি 51 আছে ... এভাবে 100 পর্যন্ত দেখব। মানে 50 থেকে 100 পর্যন্ত সব সংখ্যার জন্য total_marks অ্যারেতে সংখ্যাগুলো চেক করব। for(marks = 50; marks <= 100; marks++) { লুপের সাহায্যে প্রথমে marks-এর মান 50, তারপরে 51, এভাবে এক এক করে বাড়াব 100 পর্যন্ত।

```
count = 0; ধরে নিচ্ছি শূন্য জন 'marks' নম্বর পেয়েছে। marks-এর সব কটি মানের জন্যই প্রথমে আমরা এই কাজটি করব। এবারে total_marks অ্যারেতে দেখব যে কোনো নম্বর যদি marks-এর সমান হয়, তবে count-এর মান এক বাড়িয়ে দেব। তাহলে কোনো একটি নম্বর (marks) যতবার অ্যারেতে আছে, count-এর মান তত হবে।
```

```
for(i = 0; i < 40; i++) {  
    if(total_marks[i] == marks) {
```

```

        count++;
    }
}
printf("Marks: %d Count: %d\n", marks, count); এখানে আমরা প্রতিটি marks এবং সেটি কতবার আছে
(count) তা প্রিন্ট করে দিচ্ছি।
}
তাহলে পুরো প্রোগ্রাম লিখে ফেলি:

```

```

#include <stdio.h>
int main()
{
    int marks, i, count;
    int total_marks[] = {86, 78, 94, 68, 92, 78, 64, 62, 72, 61, 72, 66, 65, 65, 80, 72,
    for(marks = 50; marks <= 100; marks++) {
        count = 0;
        for(i = 0; i < 40; i++) {
            if(total_marks[i] == marks) {
                count++;
            }
        }
        printf("Marks: %d Count: %d\n", marks, count);
    }
    return 0;
}
প্রোগ্রাম: ৬.৮

```

তেমন কঠিন কিছু নয়। নেস্টেড ফর লুপ ব্যবহার করে সহজ-সরল সমাধান করে ফেললাম। আচ্ছা বলো তো if-এর ভেতর যে শর্তটি আমরা পরীক্ষা করছি (total_marks[i] == marks) এই কাজটি প্রোগ্রামে কতবার হয়? বাইরের লুপটি ঘুরবে 51 বার এবং প্রতিবারের জন্য ভেতরের লুপটি ঘুরবে 40 বার। তাহলে মোট $51 \times 40 = 2040$ বার।

ওপরের প্রোগ্রামটি আমরা এখন একটু অন্যভাবে লিখার চেষ্টা করব। নিচের প্রোগ্রামটি চটপট টাইপ করে ফেলো এবং রান করো:

```

#include <stdio.h>
int main()
{
    int i;
    int total_marks[] = {86, 78, 94, 68, 92, 78, 64, 62, 72, 61, 72, 66, 65, 65, 80, 72,
    int marks_count[101];
    for(i = 0; i < 101; i++) {
        marks_count[i] = 0;
    }
    for(i = 0; i < 40; i++) {
        marks_count[total_marks[i]]++;
    }
    for(i = 50; i <= 100; i++) {
        printf("Marks: %d Count: %d\n", i, marks_count[i]);
    }
    return 0;
}
প্রোগ্রাম: ৬.৯

```

এখানে আমি যেটি করেছি, একটি অতিরিক্ত অ্যারে ব্যবহার করেছি। marks_count একটি ইন্টিজার টাইপের অ্যারে এবং marks_count[n] দিয়ে আমরা বুঝব n সংখ্যাটি কতবার total_marks-এর মধ্যে আছে। নম্বর যেহেতু 0 থেকে 100-এর মধ্যে হতে পারে তাই আমরা ওই অ্যারেতে মোট 101টি সংখ্যা রাখার ব্যবস্থা করলাম। int marks_count[101]; শুরুতে যেহেতু কিছুই জানি না, তাই ধরে নিই, সব সংখ্যা শূন্য বার আছে। তাই marks_count অ্যারের সব ঘরে 0 বসিয়ে দিই:

for(i = 0; i < 101; i++) { marks_count[i] = 0; } এখন total_marks অ্যারের প্রতিটি সংখ্যার জন্য marks_count অ্যারের ওই ঘরের মান এক বাড়িয়ে দিই। for(i = 0; i < 40; i++) { marks_count[total_marks[i]]++; } বুঝতে সমস্যা হচ্ছে নাকি? একটু চিন্তা করো। যখন i-এর মান 0, তখন total_marks[i] হচ্ছে total_marks[0], অর্থাৎ 86। এখন আমাদের দরকার হচ্ছে marks_count অ্যারের ওই ঘরটার (মানে marks_count[86]) মান এক বাড়িয়ে দেওয়া। শুরুতে ছিল শূন্য, এখন হবে এক। আমরা কিন্তু সে কাজটিই করেছি marks_count[total_marks[i]]-এর মান এক বাড়িয়ে দিয়েছি marks_count[total_marks[i]]++; আসলে ব্যাপারটি এইভাবেও লেখা যেত: t_m = total_marks[i]; marks_count[t_m]++; এখনো যারা মাথা চুলকাচ্ছ তারা নিচের প্রোগ্রামটি কম্পিউটারে রান করাও। এখানে প্রতিবার marks_count[total_marks[i]]++; করার পরে marks_count অ্যারেটি আমরা এক লাইনে প্রিন্ট করেছি।

```
#include <stdio.h>
int main()
{
    int i, j;
    int total_marks[] = {6, 7, 4, 6, 9, 7, 6, 2, 4, 3, 4, 1};
    int marks_count[11];
    for(i = 0; i < 11; i++) {
        marks_count[i] = 0;
    }
    for(i = 0; i < 12; i++) {
        marks_count[total_marks[i]]++;
        for(j = 0; j <= 10; j++) {
            printf("%d ", marks_count[j]);
        }
        printf("\n");
    }
    return 0;
}
```

প্রোগ্রাম: ৬.১০

পরের অধ্যায়

Like 2.1K people like this. Be the first of your friends.

এর দ্বারা পোস্ট করা Tamim Shahriar

লেবেলসমূহ: [অ্যারে](#)

৪১টি মন্তব্য:



Abu Hasan Mahi ১৫ অক্টোবর, ২০১১ ৭:৩৪ PM

I can't find the bug of 6.6.
Can you solve it or give hints for me ?

উত্তর

উত্তরগুলি