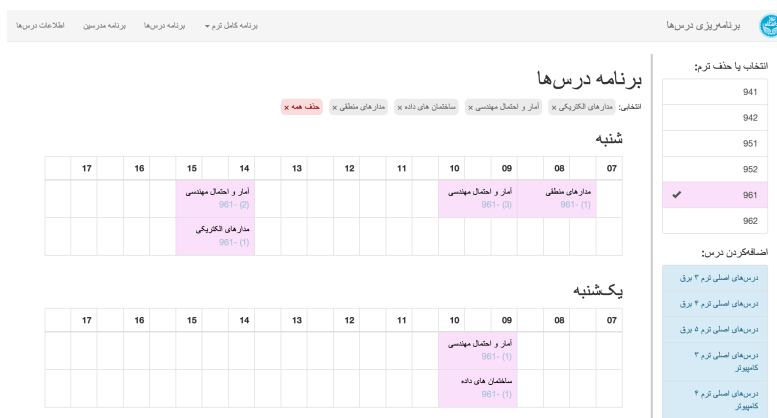


۱ مصورساز برنامه

یکی از وظایف معاونت آموزشی و اداره آموزش در دانشگاه‌ها برنامه‌ریزی دروس است. این برنامه‌ریزی کاری پیچیده و دشوار است؛ زیرا باید وقت‌های خالی استادان، وجود فضای آموزشی مناسب، عدم تداخل کلاس‌هایی که دانشجویان ترم‌های مختلف رشته‌های گوناگون قصد دارند در آن‌ها شرکت کنند، استفاده مفید از ساعات پربازده روز و چندین مؤلفه دیگر در آن در نظر گرفته شود.

معاونت آموزشی دانشکده برق و کامپیوتر دانشگاه تهران برای تسهیل برنامه‌ریزی چند ابزار کمکی را توسعه داده است که برخی اطلاعات مورد نیاز را مصور^۱ می‌کند. یکی از این ابزارها مصورساز برنامه است که اطلاعات دروس ارائه‌شده را روی برنامه هفتگی دانشکده نشان می‌دهد.



شکل ۱: مصورساز برنامه

با توجه به افزایش تعداد ورودی‌های رشته‌های مختلف دانشگاه که به پیچیده‌تر شدن فرآیند برنامه‌ریزی آموزشی منجر شده است، تصمیم بر این شد که نسخه‌ای از این برنامه به زبان ++C پیاده‌سازی شود تا بازدهی بالاتری داشته باشد و بتوان به سرعت انواع برنامه‌های رشته‌های مختلف را با آن مصورسازی کرد.

۱.۱ نحوه عملکرد برنامه

ابتدا فهرستی شامل نام انگلیسی دروس دانشکده بر اساس شناسه دروس‌ها در اختیار برنامه قرار می‌گیرد. سپس اطلاعات ارائه درس‌هایی که قرار است مصور شوند به برنامه داده می‌شوند.

از هر درس ممکن است چندین گروه متفاوت ارائه شود و هر گروه از هر درس ممکن است چند بازه زمانی در طول هفته را به خود اختصاص دهد. مثلاً درس آمار و احتمال مهندسی (با شناسه ۸۱۰۱۰۹۲) در ترم پاییز ۹۶ در سه گروه ارائه شده است. کلاس‌های گروه ۰۳ را دکتر بهرک در ساعت‌های ۰۹:۰۰ تا ۱۰:۳۰ روزهای شنبه و ۰۹:۰۰ تا ۱۰:۳۰ روزهای دوشنبه برگزار کرده است.

مصورساز ابتدا باید برای هر بازه زمانی هر گروه درسی مکان مناسب را بیابد. ممکن است در یک زمان چندین کلاس درس در دانشکده برگزار شود؛ در نتیجه، گاهی نمایش همه گروه‌های درسی در یک ردیف ممکن نیست. مصورساز برای یافتن مکان

^۱visualise

مناسب یک بازه زمانی، آن را از پایین‌ترین ردیف به برنامه یک روز هفته وارد می‌کند و تا جایی بالا می‌برد که با هیچ بازه زمانی مربوط به گروه درسی دیگری برخورد نکند. اگر پایین‌ترین ردیف روز مورد نظر در آن بازه زمانی پر بود، ردیف جدیدی به پایین آن روز اضافه می‌شود. برای فهم دقیق‌تر نحوه چینش بازه‌های زمانی **ورودی و خروجی نمونه** را بررسی کنید. توجه کنید که نحوه چینش بازه‌های زمانی با روش بیان‌شده برای هر ورودی یکتا خواهد بود و انتظار می‌رود برنامه شما دقیقاً خروجی مورد انتظار را تولید کند.

در انتها، مصورساز برنامه هر هفته را بر اساس چینش تعیین‌شده به شکل متنی تهیه می‌کند. نام انگلیسی درس بر اساس شناسه درس استخراج می‌شود و در برنامه هفتگی نمایش داده می‌شود.

۲.۱ قالب ورودی‌ها

ورودی‌های برنامه از سه بخش تشکیل می‌شود:

۱.۲.۱ فهرست نام دروس

فهرستی از نام انگلیسی دروس بر اساس شناسه درس‌ها در پرونده‌ای با قالب CSV^۲ ساده‌شده در اختیار برنامه شما قرار می‌گیرد. در خط اول این پرونده، عبارت `id, alias` درج شده است.

در هر یک از خطوط بعدی، ابتدا شناسه یک درس که یک عددی ۷رقمی است می‌آید و سپس بلافاصله بعد از یک ویرگول انگلیسی، نام انگلیسی آن درس ذکر می‌شود. تضمین می‌شود نام درس شامل فاصله^۳ یا هر نویسه فضای خالی^۴ دیگری نخواهد بود. برای مثال، یکی از خطوط بعدی ممکن است به شکل `8101119, adv-prog` باشد که درس برنامه‌نویسی پیشرفته را نشان می‌دهد.

آدرس این پرونده را به عنوان آرگومان اول خط فرمان به برنامه شما داده خواهد شد. برای آشنایی با نحوه استفاده از آرگومان‌های خط فرمان به ضمیمه **آرگومان‌های خط فرمان** مراجعه کنید.

۲.۲.۱ فهرست ارائه‌ها

فهرست گروه‌های مختلف درس‌هایی که باید مصور شوند از طریق ورودی استاندارد `stdin` در اختیار برنامه گذاشته می‌شود. هر خط ورودی شامل بخش‌های زیر می‌شود که با نویسه^۵ فاصله (با شناسه ASCII ۳۲ یا `0x20`) از هم جدا می‌شوند:

- شناسه ارائه: شناسه ارائه به‌ترتیب از شناسه درس (یک عدد ۷رقمی) و شناسه گروه (یک عدد ۲رقمی) تشکیل می‌شود که با نویسه خط (-) با شناسه ASCII ۴۵ یا `0x2D` از هم جدا شده‌اند.
- بازه‌های زمانی ارائه: هر ارائه یک یا چند بازه زمانی را به خود اختصاص می‌دهد. بازه‌های زمانی با فاصله از هم جدا می‌شوند. هر کدام از بازه‌های زمانی ساختار زیر را دارند و اجزای آن‌ها با فاصله از هم جدا می‌شوند:

◇ روز یکی از مقادیر `FRI`، `THU`، `WED`، `TUE`، `MON`، `SUN`، `SAT`

^۲مقادیر جداشده با ویرگول یا Comma-Separated Values یک قالب رایج برای پرونده‌های متنی است که در آن مقادیر با استفاده از حائل کامای انگلیسی (،) با شناسه ASCII ۴۴ یا `0x2C` از یکدیگر جدا می‌شوند. این پرونده‌ها را می‌توان با نرم‌افزارهای صفحه‌گسترده‌ای مانند `LibreOffice`، `Microsoft Excel` یا `Apple Numbers` مشاهده و ویرایش کرد.

^۳space

^۴whitespace

^۵character

◊ **زمان** شامل زمان شروع و پایان که با نویسه خط از هم جدا شده‌اند و هر یک از قالب hh:mm تبعیت می‌کنند. دقیقه (mm) یکی از دو مقدار 00 یا 30 را دارد و ساعت (hh) با یکی از مقادیر 07 تا 20 مشخص می‌شود؛ بنابراین هر یک از زمان‌های شروع و پایان از 5 نویسه تشکیل می‌شود. جداکننده ساعت و دقیقه نویسه دونقطه (:) با شناسه ASCII 5A یا 0x3A است.

مثلاً یک خط ورودی که یک ارائه درس برنامه‌نویسی پیشرفته را در ساعات ۱۰:۳۰ تا ۱۲:۰۰ روزهای یکشنبه و سه‌شنبه مشخص می‌کند ممکن است به شکل 8101119-01 SUN 10:30-12:00 TUE 10:30-12:00 باشد.

۳.۲.۱ عنوان برنامه

عنوان برنامه به عنوان آرگومان دوم خط فرمان به برنامه داده می‌شود.

۳.۱ قالب خروجی‌ها

خروجی از چندین بخش تشکیل می‌شود:

۱.۳.۱ عنوان

در ابتدای خروجی عنوان برنامه با پیشوند # نمایش داده می‌شود؛ مثلاً ممکن است Prog # سطر اول خروجی را تشکیل دهد. بعد از این سطر یک سطر خالی نمایش داده می‌شود.

۲.۳.۱ نام روز هفته

بعد از عنوان، برای هر یک از روزهای هفته نام روز با پیشوند ## نمایش داده می‌شود؛ مثلاً ممکن است Saturday ## سطر بعدی خروجی را تشکیل دهد.

نام روز به ترتیب یکی از عبارات Sunday، Monday، Tuesday، Wednesday، Thursday یا Friday خواهد بود. بعد از این سطر یک سطر خالی نمایش داده می‌شود.

۳.۳.۱ خط زمان

در ادامه در یک خط زمان‌ها با قالبی شبیه ورودی (hh:mm) می‌آیند. هر یک از ۲۸ بلوک مربوط به زمان‌ها به اندازه ۱۰ شناسه طول دارد. زمان در سمت چپ ظاهر می‌شود و ادامه بلوک با نویسه فاصله پر می‌شود. بلوک آخر استثناً حساب می‌شود و فقط به اندازه طول برجسب بلوک (۵ شناسه) طول خواهد داشت.

در خط بعد به اندازه مجموع طول بلوک‌ها نویسه زیرخط (_ با شناسه ASCII ۹۵ یا 0x5F) ظاهر خواهد شد.

۴.۳.۱ بازه‌های زمانی ارائه‌ها

در خطوط بعدی، ردیف‌های مربوط به بازه‌های زمانی ارائه‌ها قرار می‌گیرد.

هر ردیف از بازه‌های زمانی در سه سطر نمایش می‌یابد. سطر اول و سوم به حاشیه‌های افقی و گوشه‌ها اختصاص دارد و در سطر دوم حاشیه‌های عمودی و نام درس نمایش می‌یابد. حاشیه‌های افقی با نویسه خط، حاشیه‌های عمودی با نویسه خط عمودی (| با شناسه ASCII ۱۲۴ یا 0x7C) و گوشه‌ها با نویسه مثبت (+ با شناسه ASCII ۴۳ یا 0x2B) نمایش داده می‌شوند.

هر بلوک زمانی در این خطوط طولی برابر ۱۰ نویسه دارد و از ستون زیر جداکننده ساعت و دقیقه (:) همان زمان آغاز می‌شود و تا ستون قبل از جداکننده ساعت و دقیقه زمان بعدی ادامه می‌یابد. حاشیه‌های عمودی هر بازه زمانی در اولین ستون اولین بلوک و آخرین ستون آخرین بلوک آن قرار می‌گیرند. بنابراین، در ابتدای همه سطرهای مربوط به نمایش بازه‌های زمانی حداقل دو نویسه فاصله قرار خواهد گرفت.

بعد از آخرین بازه زمانی هر ردیف چیزی نمایش داده نمی‌شود.

هیچ سطر خالی‌ای از بازه‌های زمانی نمایش داده نمی‌شود.

در انتهای خروجی هر روز هفته یک سطر خالی نمایش داده می‌شود.

۵.۳.۱ نام درس

نام درس از نام انگلیسی درس، یک نویسه فاصله و شناسه گروه درس تشکیل می‌شود. شناسه گروه به شکل عدد (بدون ۰های ابتدایی) و میان دو نویسه پرانتز () با شناسه ASCII ۴۰ یا 0x28 و (با شناسه ASCII ۴۱ یا 0x29) نمایش داده می‌شود.

نام درس به شکل وسط‌چین نمایش داده می‌شود و بقیه بلوک با فاصله پر می‌شود. اگر تعداد نویسه‌های فاصله لازم برای پرکردن بلوک فرد باشد، تعداد بیشتر پیش از نام درس قرار می‌گیرد.

با توجه به توضیحات ارائه‌شده، بخشی از خروجی ممکن است به این شکل باشد:

۲۸	## Monday					
۲۹						
۳۰	07:00	07:30	08:00	08:30	09:00	09:30
۳۱	<hr/>					
۳۲	+-----++-----					
۳۳		logic (1)				prob
۳۴	+-----++-----					

در انتهای هیچ سطری نویسه فضای خالی وجود نخواهد داشت.^۶

به دلیل طولانی بودن سطرهای خروجی، نمایش خروجی در صفحه خط فرمان احتمالاً قابل فهم نیست. می‌توانید با استفاده از روش معرفی شده در ضمائم تمرین اول خروجی برنامه را در یک فایل بریزید و سپس در یک ویرایشگر متن در حالت غیرفعال بودن شکستن خط^۷ یا لفافه‌بندی کلمه^۸ آن را مشاهده کنید.

^۶no trailing whitespace

^۷line break

^۸word wrap

۴.۱ ورودی و خروجی نمونه

ورودی

```
$ ./a.out courses.csv "Prog"

8101119-01 SUN 10:30-12:00 TUE 10:30-12:00
8101347-01 SUN 09:00-10:30 TUE 09:00-10:30
8101347-02 SAT 10:30-12:00 MON 10:30-12:00
8101092-01 SUN 09:00-10:30 TUE 09:00-10:30
8101092-02 SAT 14:00-15:30 MON 14:00-15:30
8101092-03 SAT 09:00-10:30 MON 09:00-10:30
8101367-01 SAT 07:00-09:00 MON 07:00-09:00
8101045-01 SAT 13:00-16:00
8101045-02 SAT 16:00-19:00
8101045-03 SUN 13:00-16:00
8101045-04 SUN 16:00-19:00
8101045-05 MON 13:00-16:00
8101045-06 MON 16:00-19:00
8101045-07 TUE 16:00-19:00
8101480-01 SUN 14:00-15:30 TUE 14:00-15:30 SAT 12:30-14:00
```

courses.csv

```
id,alias
8120121,math-1
8120122,math-2
8120115,phys-1
8120116,phys-2
8120097,diffeq
8101092,prob-stat
8101347,fcg
8101119,adv-prog
8101523,discrete-math
8101480,discrete-struct
8101367,logic
8101045,logic-lab
```

۲ نحوه تحویل

پرونده^۹ برنامه خود را با نام A3-SID.cpp در صفحه CECM درس بارگذاری کنید که SID شماره دانشجویی شماست؛ برای مثال اگر شماره دانشجویی شما ۸۱۰۱۹۷۹۹۹ باشد، نام پرونده شما باید A3-810197999.cpp باشد.

- برنامه شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++11 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.
- کتابخانه iostream امکاناتی برای قالب‌بندی خروجی ارائه می‌دهد که در حل این تمرین به شما کمک زیادی می‌کند. برای آشنایی با این امکانات می‌توانید به بخش **Output with ostream and ios** از پایگاه **LearnCpp.com** مراجعه کنید.
- تمیزی کد، شکستن مرحله به مرحله مسأله و طراحی مناسب در کنار تولید خروجی دقیق و درست بخش مهمی از نمره شما را تعیین خواهد کرد.
- در این تمرین اجازه استفاده از مفاهیم شیء‌گرایی را ندارید.
- از صحت قالب^{۱۰} ورودی‌ها و خروجی‌های برنامه خود مطمئن شوید. توجه کنید که آزمون خودکار برنامه به تعداد و محل فاصله‌ها و خطوط خالی نیز حساس است. توصیه می‌کنیم حتماً برنامه خود را با ورودی و خروجی نمونه بیازمایید و از ابزارهایی مانند diff برای اطمینان از درستی عملکرد برنامه خود برای ورودی نمونه استفاده کنید.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.

آ آرگومان‌های خط فرمان^{۱۱}

آرگومان‌های خط فرمان آرگومان‌هایی هستند که سیستم عامل در زمان برنامه آن‌ها را به برنامه انتقال می‌دهد. سپس، برنامه می‌تواند آن‌ها را نادیده بگیرد یا از آن‌ها استفاده کند.

برای استفاده از این آرگومان‌ها، تابع main باید به شکل زیر نوشته شود:

```
int main(int argc, char *argv[])
```

دو آرگومان تابع main را می‌توان برای دسترسی به آرگومان‌های خط فرمان استفاده کرد:

- **argc** عدد صحیح؛ تعداد آرگومان‌های خط فرمان داده شده به برنامه
این مقدار حداقل برابر ۱ است؛ زیرا دستور اجرای برنامه (نام پرونده اجرایی) حتماً در زمان اجرای برنامه مورد استفاده قرار می‌گیرد و همواره به عنوان آرگومان خط فرمان شماره ۰ به برنامه داده می‌شود.
- **argv** آرایه‌ای از رشته‌های مدل زبان C^{۱۲}؛ آرگومان‌های خط فرمان داده شده به برنامه
طول این آرایه برابر argc است.

به عنوان یک مثال ساده، این برنامه را در نظر بگیرید:

^۹file

^{۱۰}format

^{۱۱}command line arguments

^{۱۲}در زبان C رشته‌ها به شکل آرایه‌ای از حروف و نویسه‌ها ذخیره می‌شوند.

```
#include <iostream>

int main(int argc, char *argv[])
{
    std::cout << "There are " << argc << " arguments:" << std::endl;

    // Loop through each argument and print its number and value
    for (int count=0; count < argc; ++count)
        std::cout << count << " " << argv[count] << std::endl;

    return 0;
}
```

اگر برنامه به شکل 100 Myfile.txt ./a.out اجرا شود، این خروجی را تولید می‌کند:

```
There are 3 arguments:
0 ./a.out
1 Myfile.txt
2 100
```

بخش [Command Line Arguments](https://www.learn-cpp.com) از پایگاه [LearnCpp.com](https://www.learn-cpp.com) نحوه کار با آرگومان‌های خط فرمان را کامل‌تر و با مثال‌های بیشتری توضیح داده است.