

## تمرین های سری دوم درس برنامه نویسی پیشرفته کامپیوتر

امیر جهانشاهی

۵ اسفند ۱۳۹۶

۱. در کدهای بخش های ۱ و ۲ بیان کنید آیا این کدها کامپایل میشوند؟ برنامه نویس چه منطقی را دنبال نمیکرده است؟ خروجی کد را نیز در صورت وجود بیان کنید. در کدهای بخش ۳ و ۴ در خط های نشان داده شده با علامت سوال مشخص کنید که این خطوط نشان دهنده چه نوع تغییری میباشند و همچنین بیان کنید خطوطی که در جلوی آن ها واژه Allowed? نوشته شده است آیا درست هستند؟ آیا اجازه استفاده از چنین نوشتاری را داریم یا خیر؟ همچنین خروجی کد را در صورت وجود بیان کنید.

(۱)

```
۱ int main()
۲ {
۳     int a{10};
۴     int* const b{&a};
۵     (*b)++;
۶     a++;
۷     std::cout << a << " " << b << " " << *b;
۸     return 0;
۹ }
```

(۲)

```
۱ int main()
۲ {
۳     const int a{10};
۴     int c{20};
۵     int d{30};
۶     const int* b{&a};
۷     b = &c;
۸     std::cout << a << " " << b << " " << *b;
۹     int* const e{&c};
۱۰    e = &d;
۱۱    std::cout << d << " " << e << " " << *e;
```

```

۱۲         return 0;
۱۳     }

```

(۳)

```

۱ char a{'a'};
۲ const char* name{"Amir Jahanshahi"};
۳ const char* p1{name}; // ?
۴ std::cout << *p1 << *(p1 + 1) << *(p1 + 2) << std::endl;
۵ p1 = &a; //Allowed?
۶ std::cout << *p1 << *(p1 + 1) << *(p1 + 2) << std::endl;
۷ p1 = name;
۸ *p1 = 'b'; //Allowed?
۹ char* p2{name}; //Allowed?

```

(۴)

```

۱ int* p1{new int[10]}; //?
۲ int* p2[10]; //?
۳ int (*p3)(int[]); //?
۴ int (*p4[10])(int [10]); //?
۵ int (*p5)[10]{new int [10][10]}; //?

```

۲. در این سوال، می‌خواهیم که یک برنامه‌ی غلط‌یاب تاییبی ساده برای واژه‌های عادی انگلیسی طراحی کنیم. به طور معمول، در زبان انگلیسی هیچ‌گاه پنج حرف بی‌صدا پشت سرهم نمی‌آیند، مگر در حالاتی که مخفف عبارتی باشند که در این صورت، با حروف بزرگ نمایش داده می‌شوند. به عنوان نمونه می‌توان به واژه‌ی HTTPS اشاره کرد.

در این سوال ورودی به صورت یک فایل txt خواهد بود. به عنوان مثال:

Falcon Heavy is a reusable super heavy-lift launch vehicle (RSHLV) designed and manufactured by SpaceX. This rshlv (previously known as the Falcon 9 Heavy) is a variant of the Falcon 9 launch vehicle and cnssts of a strengthened Falcon 9 rocket core with two additional Falcon 9 first stages as strap-on bstrs.

خروجی:

rshlv cnssts bstrs

**راهنمایی:** توضیحات کاملی در رابطه با نحوه کار با String در فصل ۷ ام کتاب موجود است. بخش Searching for any of a Set of Characters این فصل میتواند برای حل سوال مفید باشد.

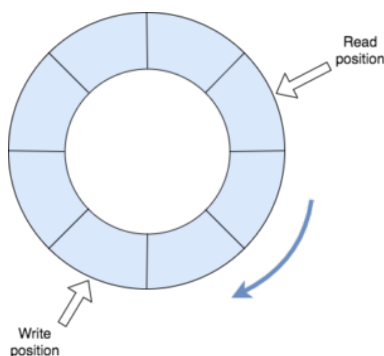
۳. در این سوال هدف نوشتن کلاسی است که یکی از انواع ساختمان داده ها به نام صف حلقوی را پیاده سازی نماید.

**آنچه باید در رابطه با صف حلقوی بدانید:** صف لیستی است که عمل افزودن داده ها درون آن از انتهای لیست و عمل حذف داده ها از ابتدای لیست انجام می شود. مثل یک صف نانوائی داده ها به ترتیب ورود پشت سر هم در صف قرار می گیرند. بنابراین اولین داده ورودی اولین داده خروجی نیز خواهد بود، این به این معنی است که شیوه عمل کرد صف براساس سیاست FIFO (First In First Out) است. ایده صف حلقوی از آنجا شکل می گیرد که اگر ما  $n$  عنصر را وارد صف کنیم و سپس آنها را یکی یکی حذف کنیم شرط پر بودن صف برقرار می ماند و این در حالی است که صف هنوز جای خالی دارد. روش هایی برای رهایی از این مشکل وجود دارند که ما در اینجا به دو مورد اشاره می کنیم:

۱- شیفت دادن خانه های انتهای صف به سمت ابتدای صف

۲- استفاده از مفهومی به این شکل که وقتی به انتهای صف رسیدیم، عمل درج را دوباره از ابتدای صف انجام بدهیم و این روند به صورت چرخشی باشد.

روش اول در بعضی موارد دارای هزینه ی بسیار بالایی خواهد بود. زمانی که تعداد خانه های ما بسیار زیاد باشند. خب مشخص هست که شیفت دادن تعداد زیادی از خانه ها آن هم به مقدار زیاد، زمان بر و پر هزینه خواهد بود. روش دوم که روند معمول و مورد استفاده در اکثر موارد است همان روش صف حلقوی است.



مطابق آنچه در شکل دیده می شود روند کاری این ساختمان داده به این صورت می باشد که براساس موقعیت این دو نشانگر نسبت به هم تعیین می کند که حافظه پر یا خالی است. حال کلاس Queue را به نحوی بنویسید که ورودی کلاس نام فایلی موجود در پوشه کلاس می باشد.

ساختار فایل به این صورت است که در خط اول اندازه حافظه صف و در خط بعدی اعداد موجود داخل صف را نمایش می دهد و در نهایت کلاس نوشته شده باید فایل main زیر را به درستی اجرا کند.

```
1 int main()
2 {
3     Queue q{"Queue.txt"};
4
5     // Display elements present in Circular Queue
6     q.displayQueue();
7
8     // Inserting elements in Circular Queue
9     q.enqueue(15.5);
10    q.enqueue(-6);
11
12    q.displayQueue();
13
14    // Deleting elements from Circular Queue
15    std::cout<<"Deleted value = "<< q.dequeue() << std::endl;
16    std::cout<<"Deleted value = "<< q.dequeue() << std::endl;
17
18    q.displayQueue();
19
20
21    return 0;
22 }
```

**توجه کنید که:** امکان نوشتن داخل صف یا اجرای تابع enqueue در زمان پر بودن صف امکان پذیر نمی باشد و می بایست با پیامی به اطلاع کاربر برسد. از طرفی پاک کردن عناصر صفی که خالی است نیز امکان پذیر نمی باشد.

۴. برنامه ای بنویسید که از کاربر تا زمانی که عدد صفر وارد کند، عدد بگیرد. بعد به روش selection sort این اعداد را مرتب کنید و در خروجی چاپ کنید.

راهنمایی:

الگوریتم **selection sort** اینگونه عمل می کند که ابتدا کوچکترین عنصر مجموعه اعداد را یافته با اولین عدد جابجا می کنیم. سپس دومین عنصر کوچکتر را یافته با دومین عدد جابجا می کنیم و این روند را برای  $n-1$  عدد اول تکرار می کند.

**توجه:** از این [لینک](#) می توانید الگوریتم های مرتب سازی را به صورت گرافیکی مشاهده کنید.



در این بخش هدف پیاده سازی دانشی است که در کارگاه گیت آموخته اید. در این راستا ابتدا در یکی از سرویس های [github](#) و یا [gitlab](#) ثبت نام نمایید.

حال یک پروژه به نام AP-HW2 به صورت Private ایجاد نمایید. برای این کار باید در قسمت New repository در زمان ساختن repository جدید حالت Private را انتخاب نمایید.

در ادامه تمرینات انجام شده را با فولدر بندی مناسب (سوال ۱ داخل فولدری به همین نام و ...) داخل این پروژه آپلود نمایید. در بخش گزارش فرآیند بارگذاری را شرح دهید و لینک تمرین را داخل گزارش ذکر نمایید.

دقت کنید که با توجه به موارد گفته شده، فایل gitignore را به نحوی طراحی کنید که تنها فایل های اصلی و make file درون git قرار داده شوند.

**توجه:** به منظور دسترسی به تمرین برای تصحیح، پس از پایان زمان تحویل تمرین پروژه را به حالت Public تغییر دهید.

جهت تحویل تمرین، هر تمرین را داخل یک فولدر بریزید که با شماره تمرین نام گذاری شده است. Q1, Q2, ... گزارش کار را به صورت PDF در فولدر اصلی تمرین ها قرار دهید. در نتیجه در فولدر اصلی فقط یک فایل گزارش موجود می باشد و تعدادی فولدر که با شماره تمرین ها نام گذاری شده است. اسم فولدر اصلی را به صورت زیر نام گذاری و سپس فشرده سازی و در قالب یک فایل ارسال کنید. توجه نمایید که از قالب فشرده سازی rar استفاده نکنید.

شماره دانشجویی-AP-HW2.zip

مهلت تحویل: تا ساعت ۲۳ سه شنبه ۱۵ اسفند ماه ۱۳۹۶

برای عضویت در روایات تلگرام برنامه نویسی پیشرفته [اینجا](#) را کلیک کنید.  
برای عضویت در کانال تلگرام برنامه نویسی پیشرفته [اینجا](#) را کلیک کنید.