دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی برق

گزارش تمرین دوم

برنامه نویسی پیشرفته

دکتر جهانشاهی

امیررضا موسوی

۹۳۲۳۰۷۲

تمرین اول :

کد شماره یک:

ـ آیا این کد کامپایل می شود ؟

بله ، از پوینتر به درستی استفاده شده است. و در واقع (b*) رفرنس و نام دیگر a است که تغییر میکند.

ـ برنامه نویس چه منطقی را دنبال می کند؟

آدرس پوینتر b ثابت است یعنی میتوانیم مقدار متغیر را با استفاده از پوینتر تغییر دهیم ولی تغییر آدرس آن امکان پذیر نیست چون مقدار آن const.است . برنامه نویس یک بار (b*) که رفرنس a است را زیاد می کند و یک بار با استفاده از خود a این کار را می کند. و با این کار هر دو a و b را تغییر می دهد و با هم برابر اند.

ـ خروجی کد:

همانطورکه در بالا گفته شد انگار به مقدار اولیه ی a دوبار اضافه شده است و خروجی آن ۱۲ استپ.

کد شماره دو:

ـ آیا این کد کامپایل می شود ؟

خیر. همانطورکه درکد این برنامه موجود است آدرس e به صورت const تعریف شده است. که این به معنی است که وقتی این پوینتر به صورت constant تعریف شده است یعنی بعد از آدرس دهی اولیه امکان دوباره آدرس دهی وجود ندارد که این یعنی نمیتواند e = &d شود و به همین دلیل کامپایل نمی شود و خطای read-only می دهد.

ـ برنامه نویس چه منطقی را دنبال می کند؟

در واقع در این برنامه ، برنامه نویس با const تعریف کردن پوینتر b باعث شده است که فقط پس از تعریف اولیه ثابت خواهد شد. به همین دلیل نمیتوان پس از مقدار دهی اولیه ، مقدار این متغیر را تغییر داد. همین باعث می شد که امکان تغییر آدرسی که پوینتر به آن اشاره می کند وجود ندارد. بنابراین b* که ربطی به تغییر آدرس ندارد بدون خطا خواهد بود ولی e = &d باعث به وجود آمدن خطا می شود.

ـ خروجی کد:

این کد به دلیل وجد خطا ، read only می دهد . ولی خروجی خط هشتم به ترتیب به جای a ، ۱۰ و به جای b آدرس متغیر C ، و به جای b* مقدار متغیر c ، ۲۰ را چاپ می کند.

کد شماره سه:

ـ ؟ ها نشان دهنده ی چه نوع متغیری می باشند و allowed ها درست هستند؟

خط سوم : const char *p1{name} پوینتری است که به یک const obj اشاره میکند.در واقع میتوان P1 را تغییر داد ولی p1* را نیمیتوان تغییر داد. بنابراین این خط از برنامه باعث خطا می شود و درست نیست.

خط پنجم : P1، p1=&a چون پوینتر است قابلیت تغییر دارد و چون متغیر a در p1 ذخیره می شود پس درست است.

خط هشتم : p1 =b* در واقع b را میخواهد در obj مربوط به p1 قرار دهدکه با توجه به کد این obj const است و خروجی برنامه خطای read only می دهد. بنابراین درست نمیباشد.

خط نهم : char *p2{name} در این قسمت چون می خواهیم دو متغیر از دو جنس متفاوت را در هم ذخیره کنیم درست نیست. زیرا میخواهیم name که از جنس const char* را در p2 که ار جنس char* است ذخیره کنیم . که درست نمیباشد.

این کدکامپایل نمیشود به دلایل بالا.

کد شماره چهار:

ـ ؟ ها نشان دهنده ی چه نوع متغیری می باشند و allowed ها درست هستند؟

خط اول : این در واقع آرایه دینامیکی است که یک بعدی می باشد و از نوع int میباشد که ۱۰ تا عضو دارد .

خط دوم : آرایه ای از جنس pointer با ۱۰ عضو است که همه ی عناصر آن از جنس pointer می باشند.

خط سوم: این در واقع یک تابع pointer است .که ورودی این تابع از نوع int است.

خط چهارم: یک تابع pointer ی است که ابعاد این تابع ۱۰ می باشد. به عنوان ورودی یک آرایه میگیرد این آرایه از نوع int و دو بعدی می باشد و یکی از ابعاد آن ۱۰ است.

خط پنجم: این نیز در واقع یک تابع pointer ی است از نوع آرایه تعریف شده است و آرایه آن دینامیک دوبعدی با ابعاد ۱۰ در ۱۰ است .

تمرین دوم

هدف از این تمرین نوشتن برنامه ای است که غلط های املایی را شناسایی کندکه در یک کلمه ۵ حرف بی صدا پشت هم آمده اند و کلمه هم حروفش کوچک است.

نکاتی در باره ی کد برنامه

در برنامه از فایل کلمه به کلمه می خوانیم و هرکلمه را در یک استریگ ذخیره می کنیم سپس چک میکنیم که کلمه ۵ حرف کوچک پشت هم نداشته باشند برای این امر از حقله for استفاده شده است و با if وجود a,e,I,o,u را در حر ۵ حرف پشت هم چک می کنیم.

در صورتی که ۵ حرف پشت هم داشتیم در مرحله بعد چک می کنیم که اگر همه حروف بزرگ بودندکلمه درست است ولی در صورت مشاهده حرف کوچک کلمه را غلط اعلام می کنیم و آن را پرینت می کنیم این کار نیز توسط یک حلقه for و یک if که مقدار ascii حروف را چک میکند انجام می دهیم.

تمرین سوم

در این سوال هدف نوشتن کلاسی است که صف حلقولی را پیاده سازی کند

نکاتی در موردکد

برای کلاس int front , int end سر و ته صف را نشان می دهند و size تعداد المان های صف را همچنین constructor باید یک استرینگ دریافت کندکه آدرس فایل است و باید یک آرایه دینامیک داشته باشیم برای صفمان. برای خواندن از فایل باید طول صف را از فایل بخونیم و سپس مقدار المان ها را به صورت استرینگ بخوانیم. با توجه به این که بین المان ها "," قرار دارد از این استفاده کرده و المان ها را جدا می کنیم و داخل صف قرار می دهیم .

با تابع enqueue یک المان به تابع اضافه می کنیم و با توجه به حلقوی بودن آن این کار را با باقیمانده end +۱ به طول صف انجام می دهیم در صورتی که صف جا داشته باشد.

با تابع dequeue یک المان را از جلوی صف حذف می کنیم در صورتی که صف خالی نباشد و با توجه به حلقوی بودن این کار را با باقیمانده front+1 به طول صف انجام میدهد

تمرین چهارم

در این تمرین هدف پیاده سازی selection sort است برای اعدادی که کاربر وارد می کند.

نکاتی در موردکد

در این کد اعداد وارد شده توسط کاربر را هر دفعه در یک استرینگ واردکرده و در یک استرینگ ذخیره می کنیم و آنها را با # از هم جدا می کنیم تا وقتی که صفر را واردکنیم

سپس این استرینگ را جستجو می کنیم و با توجه به وجود # اعداد را جداکرده و در یک آرایه به طول اعدا وارو شده که تعداد آنها را در هنگام گرفت اعداد ازکاربر داریم ذخیره می کنیم

سپس به وسیله تابع selectionsort آنه را sort میکنیم و در هر حلقه مینیمم آرایه را پیدا می کنیم و سر جایش می گذاریم که شاما ۲ حلقه for است.

Git

یک رپزوتوری درگیت ایجادکرده و در فایل تمرین های خود دستورات زیر را واردکرده و commit و push را انجام می دهیم بای فایل gitignore. هم یک فایل در آن پوشه به همین نام اضافه می کنیم و پسوند هایی که نمی خواهیم را در آن می نویسیم و سپس commit و push می کنیم.

```
git init

git add README.md

"git commit -m "first commit

git remote add origin https://github.com/AmirrezaMousavi/AP-HW2.git
```

```
git push -u origin master
```