|  |  |  |
| --- | --- | --- |
| **تمرين سري پنجم** | **مباني برنامه‌نويسي** | **نيم‌سال اول 97-96** |

***به سه نكته توجه كنيد***

الف)‌ زمان تحويل تا قبل از 22:00 روز پنجشنبه 9 آذر ماه است.

ب) پاسخ مسائل را در سايت Quera ارسال نماييد.

ج) در اين تمرينات تا حد ممكن برنامه‌هاي خود را به توابع مختلف (هر چند توابعي كوچك) بشكنيد.

1-

الف) تابع reg\_index اعداد n و m و dir را به عنوان ورودی می‌گیرد که در آن n یک عدد صحیح مثبت چند رقمی، m یک رقم و dir یا +1 یا -1 است.

**int dig\_index(int n, int m, int dir)**

خروجی این تابع یک عدد است که مشخص میکند رقم m چندمین رقم از سمت راست (اگر dir = +1 باشد) یا سمت چپ (اگر dir = -1 باشد) عدد n است. اگر رقم m در n نباشد 0 تولید می‌شود. برای مثال

**dig\_index(1000, 0, -1) 🡪 2**

**dig\_index(1000, 0, 1) 🡪1**

**dig\_index(1000, 2, 1) 🡪 0**

تعریف این تابع را بنویسید.

ب) با استفاده از این تابع، تعريف تابع زیر را بنويسيد که یک عدد را می‌گیرد و اگر عدد متقارن باشد 1 و در غیر این صورت 0 تولید می‌کند

**int sym(int n)**

ج) يك برنامه بنويسيد كه يك عدد صحيح مثبت را بگيرد و تعيين كند كه متقارن است يا نه؟ (15 نمره)

فرمت ورودي:

<n>

فرمت خروجي:

<True>/<False>

2- تابعي به اسم max\_factor بنويسيد كه عدد صحيح n را بگيرد و بزرگترين عامل اول آن را برگرداند. تابع ديگري به اسم all\_factors بنويسيد كه عدد صحيح n را بگيرد و با استفاده از تابع max\_factor همه عامل‌هاي اول عدد n را چاپ كند. تعداد دفعات تكرار هر عامل آن نيز مشخص شود.

برنامه‌اي بنويسيد كه عدد m را از كاربر بگيرد و براي همه اعداد 1 تا m تابع all\_factor را فراخواني كند. (15 نمره)

(10 نمره اضافي در صورتي كه all\_factors به صورت بازگشتي باشد).

فرمت ورودي:

<m>

فرمت خروجي:

1: <factor>^<repeat> \* <factor>^<repeat> …

2: <factor>^<repeat> \* <factor>^<repeat> …

…

m: <factor>^<repeat> \* <factor>^<repeat> …

3- تركيب از را مي‌توان با رابطه بازگشتي به صورت زير نوشت (20 نمره)

الف) شرط (شرط‌هاي) *مناسب* پايه براي اتمام رابطه بازگشتي را تعيين كنيد.

ب) تابع بازگشتي براي پياده‌سازي تركيب از  **را بنويسيد:**

**int combination(int n, int m)**

ج)‌ برنامه‌ای بنویسید که عدد n و m را از کاربر گرفته و ترکیب m از n و تعداد دفعات فراخوانی تابع combination را محاسبه کند.

4- براي حدس Goldbach طبق الگوريتم شرح داده شده در كلاس توابع زير را پيدا سازي كنيد. (20 نمره)

**int is\_prime(int n);**

**int next\_prime\_number(int n);**

**int check\_Goldbach(int n);**

با استفاده از اين توابع، تابع main را بنويسيد كه يك عدد را از كاربر بگيرد و اين حدس را براي تمام اعداد زوج بزرگتر از دو و كوچكتر از اين عدد بررسي كرده و دو مقدار عدد اول مورد نظر را چاپ كند.

فرمت ورودي:

<n>

فرمت خروجي:

Even number: <e>

Prime 1: <p>

Prime 2: <p>

Even number: <e>

Prime 1: <p>

Prime 2: <p>

…

**5- بازي حدس زدن اعداد**: (30 نمره)

در تمرين هدف اين است كه بازي ساده حدس اعداد نوشته شود. اين برنامه يه اين صورت عمل مي‌كند كه در ابتدا نوبت كامپيوتر است بنابراين برنامه عددي بين 0 تا 1000 را در نظر مي‌گيرد. كاربر سعي مي‌كند اين عدد را پيدا كند. به ازاي هر عددي كه كاربر حدس مي‌زند برنامه مشخص مي‌كند كه آيا اين حدس از عدد مد نظر بزرگتر يا كوچكتر است. زماني كه كاربر عدد مد نظر كامپيوتر را به درستي حدس زد، نوبت كاربر مي‌شود و اين بار كامپيوتر سعي مي‌كند عدد را حدس بزند و كاربر براي هر حدس كامپيوتر، بزرگي يا كوچكي آنرا مشخص مي‌كند. با حدس درست برنامه نوبت كامپيوتر مي‌شود و اين‌ بازي ادامه مي‌يابد تا جايي كه كاربر سه بار عدد 0 را پشت سر هم وارد كند. زماني كه بازي تمام شد در انتها برنامه تعداد حدس‌هاي غلطي كه كاربر و كامپيوتر در كل بازي داشته‌اند را چاپ مي‌كند.

برنامه خود را طوري بنويسيد كه **به طور متوسط كمترين مقايسه‌ها** را داشته باشد.

**اختياري**. برنامه كامپيوتري در بازي تقلب مي‌كند!!!. برنامه هر بار جواب درست را در خصوص كوچك يا بزرگ بودن حدس كاربر نمي‌دهد! بلكه يك مقدار x وجود دارد كه نشان دهنده احتمال تقلب كامپيوتر است. مثلا اگر x=0 باشد يعني هيچ تقلبي در كار نيست. x=0.5 به معني غلط بودن جواب‌هاي كامپيوتر در 50% موارد است. برنامه بازي را به نحوي تغيير دهيد كه در ابتداي بازي مقدار x را از كاربر بگيرد و در ادامه به همان ميزان در بازي تقلب كند.