۱- خروجی کد زیر چیست؟ قدم به قدم علت پاسخ را بنویسید

Consider the following C program that contains global, local, and static variables:

```c
#include <stdio.h>

int i = 10;    /* global variable */
int j = 1;     /* global variable */

/* Forward function declarations */
void swap(int, int);
int f1(int);
void f2(int, int *);

int main()
{
   {
      int i = 1;
      int j = 10;

      swap(i,j); printf("i = %d  j = %d\n", i,j);
      j = f1(i++); printf("i = %d  j = %d\n", i,j);
      f2(f1(i),&j); printf("i = %d  j = %d\n", i,j);
   }

   j = f1(i/2); printf("i = %d  j = %d\n", i,j);
   f2(f1(i),&j); printf("i = %d  j = %d\n", i,j);

   return 0;
}

void swap(int a, int b)
{
   int temp;
   temp = b; b = a; a = temp;
}

int f1(int x)
{
   static int j = 5;
   i++;
   j += x;
   return j;
}

void f2(int x, int *p)
{
   *p += x;
   *p = (*p > 20) ?  20 : *p++;
}
```

۲- تابعی تحت عنوان MeanNums بنویسید. این تابع در هر بار فراخوانی، یک عدد را از کاربر میگیرد و با بقیه اعداد گرفته شده قبلی جمع و میانگین آن را محاسبه میکند. برای این منظور در تابع اصلی از کاربر عدد دریافت کنید، پس از وارد شدن هر عدد میانگین اعداد وارد شده تا این عدد را توسط تابع MeanNums محاسبه و نمایش دهد. (عدد 100- میتواند خاتمه این روند باشد) (استفاده از متغیر ایستا)

۳- برنامه زیر را در دو فایل مشخص شده در نظر بگیرید. اسکوپ متغیر x و y را مشخص کنید و کدام از دسترسی ها به آن صحیح و کدام خطای کامپایلری است.

File1.c

```
extern int y;
void f4()
   {
      extern int x;
      x = 4;
      y++;
   }

   void f3()
   {
      x = 4;
   }

   extern int x;

   void f2()
   {
      x = 4;
   }
```

File2.c

```
void f1()
   {
      x = 4;
      y = 20;

   }

   int x;
   static int y;
   void f2()
   {
      x = 4;
      y = 30;
   }
```

۴- برنامه ای بنویسید که کاراکتر ورودی را توسط تابع ToggleChar، در صورتیکه که حروف لاتین بزرگ است به حرف لاتین کوچک و بر عکس تبدیل کند. (ترجیحا کل این فرایند توسط عملگر :? انجام شود)

۵- تابعی تحت عنوان chrTOint بنویسید که با دریافت یک کارکتر، توسط یک دستور if تک خطی (عملگر :?) یک کاراکتر ’0‘ تا ’9‘ را به عدد صحیح متناظر آن تبدیل کنید و در غیر این صورت عدد 1- را برگرداند.

۶- برنامه ای بنویسید که شامل حلقه for با تعداد تکرار 10^8 باشد. یک بار شمارنده حلقه به صورت int تعریف و در دور بعدی به صورت register int تعریف شود. زمان اجرای این دو حلقه را محاسبه کنید و چاپ کنید که تقریبا چند برابر همدیگر طول میکشد. برای محاسبه زمان اجرا باید از کتابخانه time.h استفاده کنید و مشابه زیر عمل کنید:

```
clock_t begin = clock();

/* here, do your time-consuming job */

clock_t end = clock();
```

```
double time_spent = (double)(end - begin) / CLOCKS_PER_SEC;
```