



دانشکدهی مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران مبانی کامپیوتر و برنامهنویسی



عنوان: آزمایشگاه ۳ استاد : دکتر مرادی

نیمسال دوم ۹۸–۹۹

۱. انجام دهید!

و در خروجی x از حلقه x برنامه ای بنویسید که عدد صحیح x را از کاربر گرفته و حاصل x را محاسبه کند و در خروجی چاپ کند.

برنامه ای که در قسمت قبل نوشتید را به صورت یک تابع به نام fact دربیاورید. این تابع عدد صحیح \mathbf{x} را به عنوان آرگومان \mathbf{x} گرفته، سپس حاصل \mathbf{x} را حساب کرده و به عنوان خروجی بازمی گرداند.

- برنامه ای در تابع main بنویسید که عدد n را از کاربر بگیرد و مقدار صحیح زیر را با استفاده از تابع n حساب کرده و در خروجی چاپ کند:

$$\sum_{i=0}^{n} (-1)^{i} (i!)$$

- اکنون اظهار ٔ تابع fact را در یک header file به نام fact.h قرار دهید و بدنه ی آن را درون یک فایل cpp با نام fact.cpp قرار دهید. دقت کنید که fact.h را در ابتدای include ،fact.cpp کرده باشید. محتوای دو فایل fact.cpp و fact.cpp در زیر آمده است:

```
fact.h
int fact(int x);

fact.cpp
#include "fact.h"
int fact(int x) {
    /* Your Code Goes here. */
}
```

حال با توضیحات بالا قسمت قبل را دوباره انجام دهید:

```
main.cpp
#include "fact.h"
int main() {
    /* Your Code Goes here. */
}
```

¹ argument

² declaration

```
کد نوشته شده جهت محاسبه فاکتوریل:
#include <stdio.h>
int main(){
       int x, a = 1;
       scanf_s("%d", &x);
       int b = x;
       for (int i = 0; i < x; i++){
              a *= b;
              b--;
       printf("%d! = %d", x, a);
}
                                                              تابع fact جهت محاسبه فاكتوريل:
#include <stdio.h>
int main(){
       int x;
       scanf_s("%d", &x);
       int a = fact(x);
       printf("%d! = %d", x, a);
int fact(int x){
       int a = 1;
       int b = x;
       for (int i = 0; i < x; i++){
              a *= b;
              b--;
       return a;
}
                                                    محاسبه عبارت گفته شده با استفاده از تابع fact:
#include <stdio.h>
int main(){
       int x, result = 0;
       scanf_s("%d", &x);
       for (int i = 0; i <= x; i++){</pre>
              int a = fact(i);
              int b = i \% 2 + 1;
              if (b == 2) {
                     b = -1;
              result += a * b;
       }
       printf("%d", result);
int fact(int x){
       int a = 1;
       int b = x;
       for (int i = 0; i < x; i++){
              a *= b;
              b--;
       return a;
```

: fact برای تابع header file

```
fact.h
int fact(int x);
fact.cpp
#include "fact.h"
int fact(int x) {
      int a = 1;
      int b = x;
      for (int i = 0; i < x; i++) {</pre>
            a *= b;
            b--;
      return a;
}
main.cpp
#include <stdio.h>
#include "fact.h"
int main() {
      int x, result = 0;
      scanf_s("%d", &x);
      for (int i = 0; i <= x; i++) {
            int a = fact(i);
            int b = i \% 2 + 1;
            if (b == 2) {
                  b = -1;
            result += a * b;
      printf("%d", result);
}
```

۲. انجام دهید!

- در این قسمت قرار است تا تابع فیبوناچی را حساب کنید. یادآوری می شود که: F(n) = F(n-1) + F(n-2); F(1) = F(2) = 1; ... long f ibonacci (long f) f are f and f are f are f are f are f and f are f are f and f are f are f and f are f and f are f are f and f are f are f and f are f and f are f are f and f are f are f and f are f and f are f are f and f are f are f and f are f and f are f are f and f are f and f are f and f are f are f and f are f are f and f are f and f are f are f are f and f are f are f and f are f and f are f are f and f are f are f and f are f and f are f are f and f are f are f and f are f and f are f are f and f are f are f and f are f and f are f are f and f are f are f and f are f and f are f are f and f are f and f are f and f are f are f are f are f and f are f are f are f are f and f are f are f and f are f are f and f are f are f are f are f and f are f are f and f are f are f are f are f and f a

- تابع پیادهسازی شده در بالا را به گونه ای تغییر دهید تا در صورتی که کاربر عدد نامناسبی را وارد نمود (برای مثال

// your code goes here

}

قسمت ۲: : موارد خواسته شده را انجام داده و نتایج به دست آمده را در کادر زیر بنویسید.

اعداد منفی) مقدار ۰ را به عنوان خطا بازگرداند.

```
کد خواسته شده به همراه اعلام خطا در صورت وارد کردن عدد نامناسب:
#include <stdio.h>
int main() {
      long fibonacci(long x);
      long x;
      scanf_s("%d", &x);
      int a = fibonacci(x);
      if (a == 0)
            printf("Error: enter n>2");
      else
            printf("f(%d)=%d", x, a);
long fibonacci(long x) {
      if (x < 3)
            return 0;
      else{
            long a = 1, b = 1, c;
            for (long i = 2; i < x; i++){
                   c = a + b;
                   a = b;
                   b = c;
            return c;}
```

۳. فکر کنید!

۱. به قطعه کد زیر نگاه کنید. به نظر شما بعد از اجرای برنامه چه اتفاقی میافتد؟

```
long fib(long n)
{
   if (n <= 2)
      return 1;
   return fib(n - 1) + fib(n - 2);
}</pre>
```

قسمت ۲: نتیجه را در کادر زیر توضیح دهید.

کد بالا به زیبایی تمام جمله nام دنباله فیبوناچی را حساب میکند.

در واقع عدد n را از كاربر ميگيرد و جمله nام را برابر مجموع دو جمله قبلى خود قرار ميدهد. و با استفاده از همين تابع جمله (n-1)ام و جمله (n-2)ام را بدست مىآورد و با استفاده از همين تابع جملات قبلى آنها را محاسبه ميكند تا به جمله اول و دوم ميرسد و آنها را برابر ۱ قرار ميدهد.حال حاصل جمله nام مشخص است و آنرا return ميكند.

۴. انجام دهید!

۱. یک فایل جدید ایجاد کرده و برنامه زیر را در آن بنویسید و سعی کنید آن را کامپایل نمایید.

```
#include <stdio.h>
int main() {
      int z = 4;
      if (z - 4) {
             int I = 1;
             z = z + I;
      }
      else {
             int I = 1;
             z = I + 1;
                    int I = 1;
                    z = I + 1;
             z = I;
      z = z + I;
      do {
             int I = 0;
             z = I + 1;
             i++;
      } while (I < 15);</pre>
      return 0;
}
```

قسمت ۴:

- ۱) چرا این برنامه کامپایل نمی شود؟ در کادر زیر توضیح دهید.
- کلیه ی خطاهای کامپایلی این کد را با ذکر شماره خط بیان کرده و آن ها را رفع کنید. نتایج به دست آمده را در کادر زیر بنویسید.

کلیه خطاهای بوجود آمده در هنگام کامپایل ۲۲ تا میباشد که ۱۷ تای آن بدلیل خط ششم و عبارت if(z-4) میباشد که با عبارت (z-4) جایگزین میشود.

در خط ۱۹ و ۲۴ '۱' مقدار دهی و معرفی نشده. و در خط ۲۳ 'i' مقدار دهی و معرفی نشده است. با تغییر ۱با i در خط ۲۴. و معرفی مقدار i و ۱ در ابتدا و مقدار دهی کردن آنها میتوان سایر خطاها را رفع کرد.

علت خطاها این بود ما ۱ رو در حلقه if و do تعریف و مقدار دهی کردیم ولی در بیرون از آنها نیز از این متغییر استفاده میکنیم. برای حل این مشکل کافیست ۱ را در همان ابتدا معرفی و مقدار دهی کنیم تا بتوانیم در همه جا از آنها استفاده کنیم.

۵. انجام دهید!

قسمت ۵: برنامه زیر را اجرا کنید و سپس مقدار خروجی برنامه را نوشته و هر مقدار چاپ شده را توجیه کنید. این نتایج را در کادر تعبیه شده بنویسید.

```
int g(int y) {
    int j = 2;
    y = j * 3;
    return y;
}

int f(int x) {
    return g(x);
}

int main()
{
    int a = 10;
    printf("%d", f(a));
    return 0;
}
```

تابع g به صورتی نوشته شده که هر عددی مانند x را از کاربر بگیرد آنگاه : g(x)=6 تابع g را فرا میخواند. پس : g(x)=6 f(x)=g(x)=6 تابع g را فرا میخواند. پس : g(x)=6 قرار داده و مقدار g(x)=6 را که برابر با g(x)=6 خواهد بود پرینت میکند.

6 . انجام دهید! (امتیازی)

۱) خروجی این برنامه های زیر چیست ؟ دلیل خود را به صورت کامل تشریح کنید .

```
#include<stdio.h>
int function();
main()
{
    int i;
    i = function();
    printf("%d", i);
    return 0;
}
function()
{
    int a;
    a = 250;
}
```

```
#include<stdio.h>

void dynamic(int s, ...)
{
    printf("%d ", s);
}

int main()
{
    dynamic(2, 4, 6, 8);
    dynamic(3, 6, 9);
    return 0;
}
```

```
#include<stdio.h>
void fun(int x)
{
    if(x > 0)
    {
        fun(--x);
        printf("%d\t", x);
        fun(--x);
    }
}
int main()
{
    int a = 4;
    fun(a);
    getchar();
    return 0;
}
```

خروجی برنامه سمت چپ بالا عددی ۸ رقمی میباشد که مربوط به بیت های حافظه است. خروجی برنامه سمت راست بالا "3 2" میباشد.

خروجی برنامه پایین به صورت 1 0 3 0 2 1 0 میباشد.

موفق باشيد