
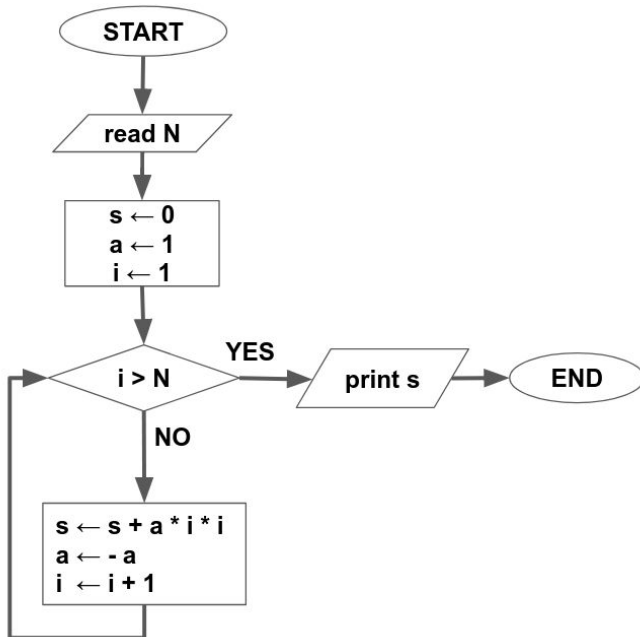


مبانی برنامه سازی - پاییز ۹۶	دکتر نیک انجام دکتر نصیحت کن	 دانشگاه صنعتی خواجه نصیرالدین طوسی K. N. TOOSI UNIVERSITY OF TECHNOLOGY
آذر ۹۶	شماره دانشجویی	نام:

قسمت اول - سوالات

کتبی

سوال ۱- (۷ نمره) فلوجارت روبرو چه کاری انجام می دهد؟ (با مثال توضیح دهید)



سوال ۲- (۸ نمره) تابع بازگشتی روبرو با گرفتن یک عدد صحیح n به عنوان آرگومان چه چیزی را به عنوان مقدار بازگشتی بر می گرداند؟ عملکرد تابع برای اعداد منفی چگونه خواهد بود؟

```

int f(int n) {
  if (n < 0)
    return f(-n);

  if (n < 10)
    return n;

  return f(n/10);
}
  
```

سوال ۳- (۵ نمره) یک حلقه while بنویسید که معادل حلقه for باشد.

```
for (int i = 1; i <= m; i+= 2) {  
  
    scanf("%d", &p);  
  
    if (i == p)  
        i++;  
}
```

سوال ۴ (۱۰ نمره) بعد از اجرای برنامه زیر چه چیزی در خروجی چاپ می شود؟ خروجی را در ستون سمت راست بنویسید.

```
#include <stdio.h>  
  
int t = 4;  
int f1();  
void f2();  
  
void main() {  
    int t = 2;  
    printf("%d\n", t);  
    t = f1();  
    {  
        int t = 3;  
        printf("%d\n", t);  
        t += 3;  
    }  
    t--;  
    printf("%d\n", t);  
    f2();  
}  
  
int f1() {  
    printf("%d\n", t);  
    t--;  
    return t;  
}  
  
void f2() {  
    printf("%d\n", t);  
}
```

قسمت دوم - سوالات عملی

سوال ۱- (۲۰ نمره) برنامه‌ای بنویسید که مثلث فلویید را رسم کند. مثلث فلویید در زیر نشان داده شده است. برنامه باید عدد صحیح n را از کابر بگیرد و n سطر اول مثلث فلویید را چاپ کند. برنامه باید دقیقاً با نام prog1.c (با حروف کوچک) ذخیره شود.

مثال:

ورودی:

4

خروجی:

```
1
2  3
4  5  6
7  8  9  10
```

ورودی:

3

خروجی:

```
1
2  3
4  5  6
```

ورودی:

1

خروجی:

1

سوال ۲- (۲۵ نمره) فرض کنید تابع J به صورت زیر تعریف شود:

$$\begin{aligned} J(1) &= 1 \\ J(2k) &= 4 J(k) - 2 \\ J(2k+1) &= 4 - J(k) \end{aligned}$$

تابعی به صورت بازگشتی بنویسید که عدد n را به عنوان پارامتر بگیرد و $J(n)$ را به عنوان مقدار برگشتی برگرداند. برنامه شما باید شامل کد زیر باشد. فقط اجازه دارید محتویات تابع J را عوض کنید و J حتما باید به صورت بازگشتی تعریف شود. برنامه باید دقیقا با نام prog2.c ذخیره گردد.

```
#include <stdio.h>

int J(int);

int main() {
    int n;

    scanf("%d", &n);

    printf("%d\n", J(n));

    return 0;
}

int J(int n) {
    // write your code here!
}
```

سوال ۳- (۳۵ نمره) تابع \exp در کتابخانه استاندارد C، یک عدد x از نوع double را به عنوان پارامتر می گیرد و e^x را به عنوان خروجی بر می گرداند. با استفاده از این تابع می خواهیم لگاریتم طبیعی اعداد مثبت را پیدا کنیم. برای پیدا کردن لگاریتم یک عدد مثبت دلخواه a کافی است معادله زیر را حل کنیم:

$$\exp(x) - a = 0$$

یا به عبارتی ریشه تابع ریاضی $f(x) = e^x - a$ را بدست آوریم. برای بدست آوردن ریشه این تابع می خواهیم از روش نیوتون استفاده کنیم. در این روش از یک نقطه اولیه x_0 شروع می کنیم و طبق معادله زیر نقاط بعدی را حساب می کنیم:

$$x_{k+1} = x_k - f(x_k) / f'(x_k)$$

که f' مشتق تابع f است. دنباله این اعداد به ریشه $f(x)$ میل خواهد کرد.

برنامه ای بنویسید که عدد مثبت a را بگیرد و لگاریتم طبیعی a را به روش بالا حساب نموده و تا ۱۰ رقم اعشار چاپ کند. روش نیوتون باید زمانی متوقف شود که

$$|x_k - x_{k-1}| < 0.0000001$$

و در این صورت جمله آخر (x_k در بالا) باید چاپ شود.

برنامه حتما باید با روش نیوتون نوشته شود. استفاده از تابع \log از کتابخانه استاندارد C مجاز نمی باشد. برای محاسبه قدر مطلق اعداد اعشاری می توانید از `fabs` استفاده کنید. مقدار اولیه x (یا همان x_0 در دنباله بالا) را برابر a قرار دهید. متغیرهای `floating point` را به صورت `double` تعریف کنید. الگوریتم باید بدون استفاده از آرایه ها نوشته شود. برنامه باید شامل کد زیر باشد و دقیقا با نام `prog3.c` ذخیره گردد.

```
#include <stdio.h>
#include <math.h>

int main() {
    double a,x;
    // define variables
    scanf("%lf", &a);

    // write your code here

    printf("%.10f\n", x);
}
```

مثال:

ورودی:

2

خروجی:

0.6931471806

ورودی:

1

خروجی:

0.0000000000

ورودی:

0.1

خروجی:

-2.3025850930

سوال جایزه ای (۲۰ نمره)

خرد کردن پول: در این برنامه می خواهیم مبلغی را به عنوان ورودی از کاربر بگیریم و تعیین کنیم که آیا با سکه هایی که در اختیار داریم می توانیم مبلغ را خرد کنیم یا نه. در خط اول ورودی کل مبلغی را که در اختیار داریم و در در خطهای بعدی لیستی از سکه هایی که پول را باید با آنها خرد کنیم (از بزرگ به کوچک) وارد می شود. در پایان عدد 0 به معنای پایان ورودی هاست. برای مثال ورودی زیر را در نظر بگیرید.

5250
500
200
50
0

در اینها مبلغ 5250 تومان در اختیار داریم و می خواهیم آن را با سکه ۵۰۰ تومانی، ۲۰۰ تومانی و ۵۰ تومانی خرد کنیم.

الگوریتم خرد کردن بسیار ساده است. از اولین سکه (در این مثال ۵۰۰ تومانی) شروع می کنیم و تا می توانیم مبلغ سکه را از مبلغ مذکور کم می کنیم تا جایی که دیگر امکان کم کردن وجود نداشته باشد. (در این مثال پس از این مرحله ۲۵۰ تومان از مبلغ باقی می ماند). پس از آن سراغ سکه بعدی می رویم و همین کار را روی مبلغ باقی مانده انجام می دهیم تا امکان کم کردن وجود نداشته باشد و همینطور این کار را تا آخرین سکه انجام می دهیم. اگر در پایان مبلغ باقیمانده برابر صفر باشد عمل خرد کردن با موفقیت انجام شده و برنامه باید کلمه YES را (با حروف بزرگ) چاپ کند. برای مثال خروجی برنامه با ورودی بالا

YES

خواهد بود. زیرا

$$5250 = 10 * 500 + 1 * 200 + 1 * 50.$$

اگر عمل خرد کردن با الگوریتم فوق موفقیت آمیز نبود و در نهایت مبلغی باقی ماند، برنامه باید کلمه NO و به دنبال آن مبلغ باقی مانده را چاپ کند. برای مثال با ورودی زیر

5254
500
40
8
0

خروجی به این صورت خواهد بود

NO 6

زیرا

$$5254 = 10 * 500 + 6 * 40 + 1 * 8 + 6$$

دقت کنید که لازم نیست سکه ها، سکه های رایج کشور باشند. در مثال بالا سکه های ۴۰ و ۸ تومانی وجود داشت.

- برنامه باید بدون استفاده از آرایه ها نوشته شود در غیر این صورت نمره ای تعلق نخواهد گرفت.
- برنامه باید دقیقاً با نام prog4.c ذخیره شود.