



دستورات کنترلی



ساختارهای کنترل

- همیشه برنامه بصورت ترتیبی اجرا می شود مگر اینکه اتفاق خاصی رخ دهد.
- در این حالت ما از ساختارهای کنترل برنامه استفاده می کنیم.
- ساختارهای کنترلی دو نوع عملکرد مفید را در اختیار ما قرار میدهند:
 - انتخاب بین گزینه های مختلف بر حسب شرایط موجود
 - تکرار یا حلقه

انتخاب

- یک ساختار کنترلی انتخاب، برای انتخاب بین گزینه های متفاوت استفاده می گردد.
- باید یک **شرط** وجود داشته باشد که بر اساس آن گزینه مورد نظر انتخاب شود.
- ساختارهای کنترلی انتخاب در C :
 - if if/else if/else if switch

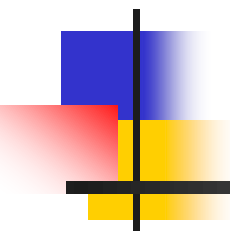


ساختارهای کنترل تکرار (Repetition Control Structures)

■ ساختارهای کنترلی **تکرار** به ما امکان می دهند که قسمتی از برنامه را به تعداد دلخواه اجرا کنیم.

■ C دارای ساختارهای تکرار زیر است:

while for do-while



ساختار انتخاب

دستور if

دستور if

- ساختار کنترلی if پس از بررسی صحت یک شرط اجازه انجام یک دستور العمل یا یک بلوک از دستورات را صادر مینماید.
- نحوه استفاده از دستور if به شکل زیر است:

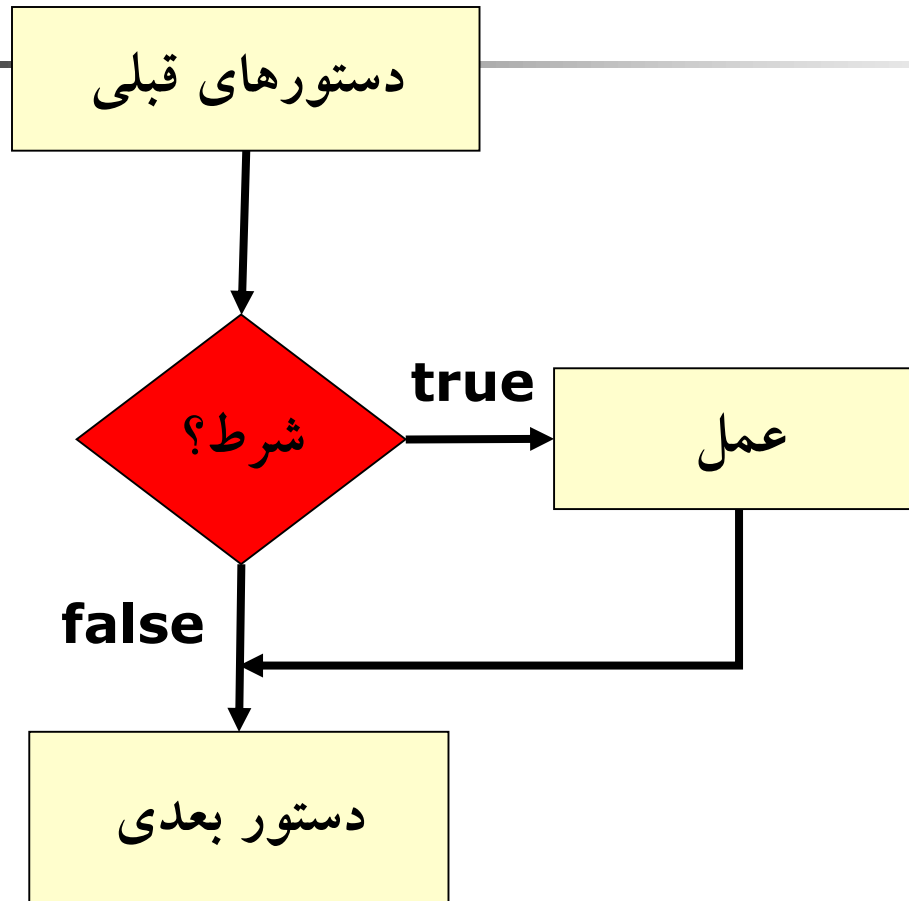
if (**<عبارت منطقی>**)

{

< if بدنه >

}

فلوچارت مربوط به دستور if



عملگرهای شرطی

■ همانطور که قبلاً آشنا شدیم این عملگرها خروجی منطقی دارند :

■ عملگرهای مقایسه ای:

■ $>$ بزرگتر

■ $>=$ بزرگتر مساوی

■ $<$ کوچکتر

■ $<=$ کوچکتر مساوی

■ عملگرهای تساوی:

■ $==$ مساوی

■ $!=$ مخالف

مثال:

برنامه ای بنویسید که کاراکتری را از کاربر گرفته و اگر برابر 'y' بود یک پیغام نشان دهد

```
int main()
{
    char ch;
    printf ( "Enter a character:" );
    ch = getch();
    if (ch == 'y' )
    {
        printf ( "\n You typed y." );
        printf ( "\n Not some other letter." );
    }
    getch();
    return 0;
}
```



عملگرهای منطقی

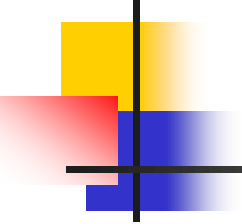
■ این عملگرها بر روی اپرندهای منطقی عمل میکنند و یک مقدار منطقی را برمیگردانند:

■ AND , OR , NOT

x	y	x AND y	x OR y	NOT x
x	y	x && y	x y	!x
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

عملگرهای شرطی و عملگرهای منطقی

- ورودی عملگرهای شرطی، عبارات ریاضی هستند
- خروجی عملگرهای شرطی یک مقدار منطقی است
- ورودی عملگرهای منطقی، عبارات شرطی هستند
- خروجی عملگرهای منطقی یک مقدار منطقی است



■ با استفاده از عملگرهای منطقی میتوان عبارات شرطی مورد نظر را تولید نمود.

■ مثال: عبارت شرطی، X بزرگتر از Y و Y مخالف صفر و X عدد مثبت باشد به شکل زیر نوشته میشود

■ $(x > y) \ \&\& \ (y \neq 0) \ \&\& \ (x > 0)$

برنامه ای بنویسید که تعداد واحد اخذ شده توسط کاربر و معدل را گرفته و اگر تعداد واحدها بیشتر از ۱۴ و معدل او بالاتر از ۱۸ بود به او پیام تبریک بگوید.

```
void main()  
{  
    float units,average;  
    printf ( "Enter Units , Average\n:" );  
    scanf ( "%f %f", &units, &average );  
  
    if (units>14 && average >18)  
    {  
        printf ( "\n You are the best." );  
        printf ( "\n You will be a good programmer!!!!." );  
    }  
    getch();  
}
```



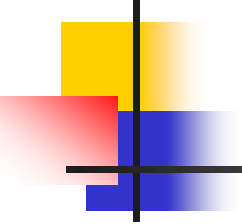
عبارت شرطی

■ در ساختارهای انتخاب، عبارت شرطی یک مقدار منطقی (Boolean) دارد که درست یا غلط است

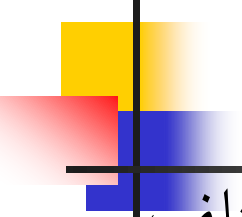
■ در C:

■ مقدار $\bullet$ برابر غلط است

■ هر مقدار دیگری برابر درست است

- 
- در صورتی که بدنه دستور `if` تنها یک دستور داشته باشد میتوان آکولادها را ننوشت
 - مثال:

```
if (3<4)  
    printf( "I am true!\n");
```



دستورات if تودرتو

- میتوان دستورات if را پشت سر هم برای بررسی شرایط مختلف استفاده نمود. یعنی یک دستور if بخشی از بدنه دستور if دیگری باشد

```
char main()
{
    if (getche() == 'n' )
    {
        if (getche() == 'o' )
        {
            printf ( "\n You typed no." );
        }
    }
    return getch();
}
```

```
char main()
{
    if (getche() == 'n' )
        if (getche() == 'o' )
            printf ( "\n You typed no." );
    return getch();
}
```



اشتباهات معمول

■ اغلب عملگر انتساب (=) با عملگر تساوی (==) اشتباه گرفته میشود.

■ در عبارت زیر چه ایرادی وجود دارد:

```
if (grade=100)
    printf( "your grade is perfect!\n" );
```



ساختار کتتری if-else

- در دستور if اگر شرط درست باشد، یک دستور یا مجموعه‌ای از دستورات اجرا میشود.
- ساختار کتتری if else اجازه می‌دهد که در صورت صدق نکردن یا غلط بودن شرط، عمل دیگری انجام گردد.

if (**<عبارت شرطی>**)

{

<بدنه if>

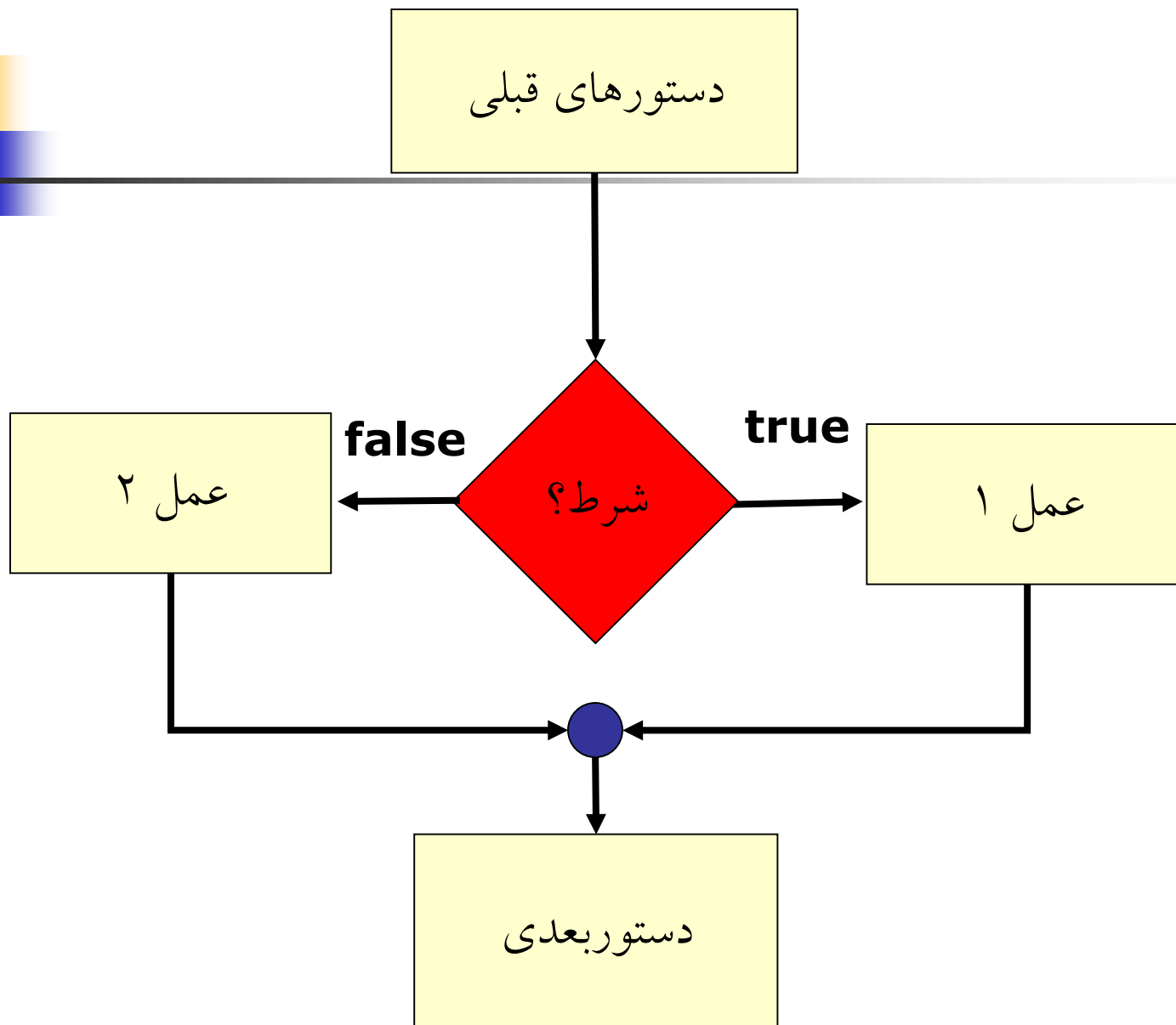
}

else

{

<بدنه else>

}



```
char main()
{
    char ch;
    printf ( "Enter a character:" );
    ch = getch();
    if (ch == 'y' )
    {
        printf ( "\n You typed y." );
    }
    else
    {
        printf ( "\n You did not type y." );
    }
    return getch();
}
```



دستورات if-else های متوالی (if-else if)

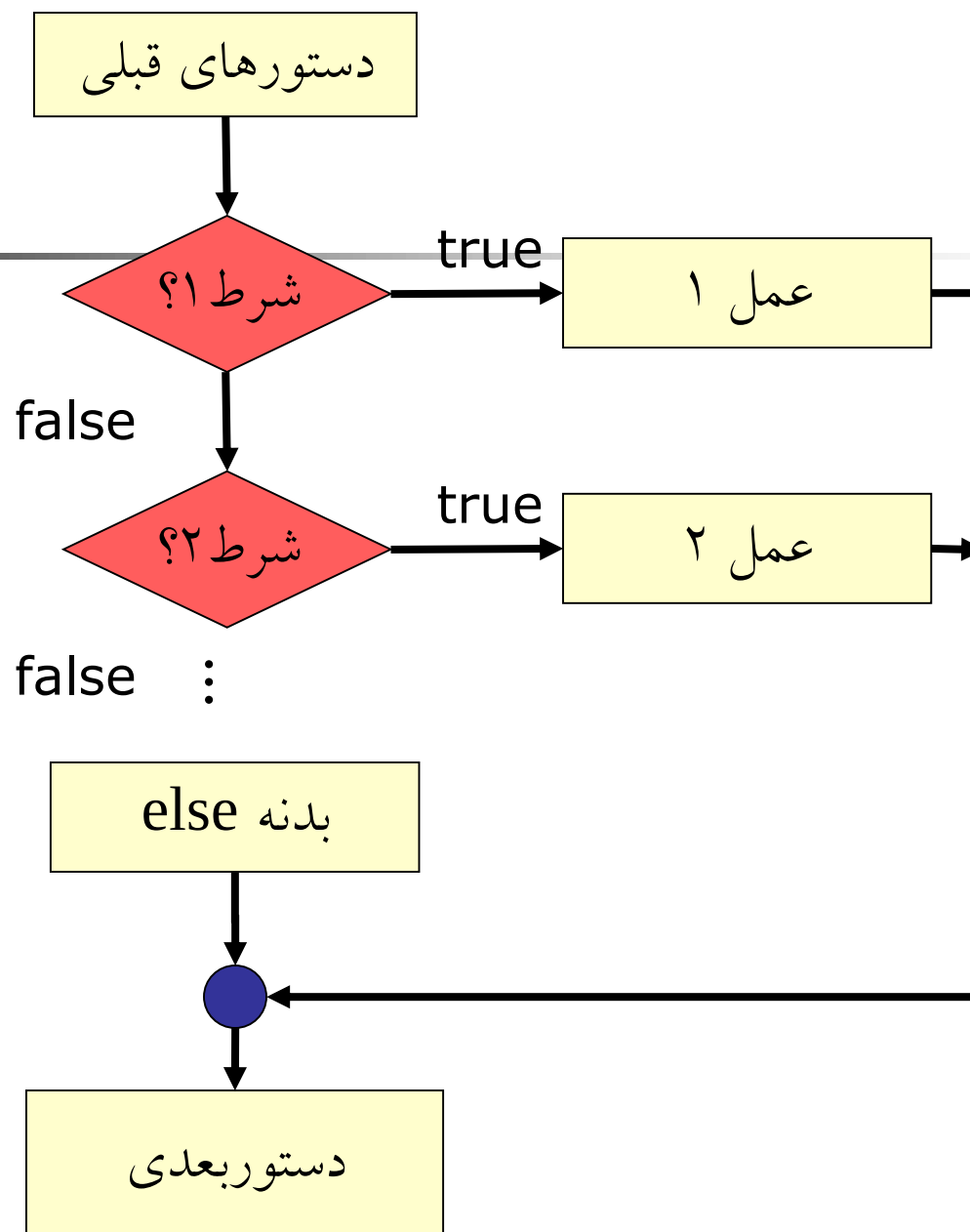
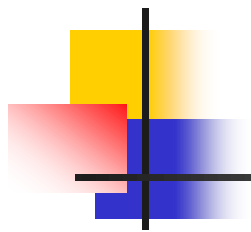
- تا کنون با استفاده از دستور **if-else** ساختاری پیاده سازی شد که در صورت صحت شرط یک سری دستورات و در صورت عدم صحت شرط مجموعه ای دیگر از دستورات انجام میشدند
- در بسیاری از کاربردها لازم است که شرایط مختلفی بررسی شوند و با توجه به صحت هر یک از شرطها عملیات مناسب انجام پذیرد
- همچنین اینکه در صورت صحت یک شرط دیگر لازم نیست شرطهای دیگر بررسی شوند



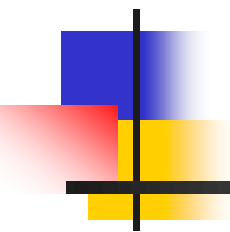
دستورات if-else های متوالی

■ بدین ترتیب میتوان از ساختار if-else متوالی استفاده نمود

```
if (<عبارت شرطی ۱> )  
{  
    <بدنه شرط ۱>  
}  
else if (<عبارت شرطی ۲> )  
{  
    <بدنه شرط ۲>  
}...  
else  
{  
    <بدنه else>  
}
```



```
char main()
{
    char ch;
    float grade;
    printf ( "Enter grade:" );
    scanf ( "%f" , &grade);
    if (grade > 18.0)
        ch = 'A' ;
    else if (grade > 16.0)
        ch = 'B' ;
    else if (grade > 14.0)
        ch = 'C' ;
    else if (grade > 10.0)
        ch = 'E' ;
    else
        ch = 'F' ;
    printf ( "\n Your Level is: %c" , ch );
    return getch();
}
```



تمرین مربوط به کاربرد if

زمان تحویل: جلسه آینده

تمرین الف

■ برنامه حل معادله درجه دو را به شکل زیر اصلاح نمایید:

■ در صورتی که دلتا منفی بود پیغام مناسب داده شود.

■ در صورتی که دلتا برابر با صفر بود وجود ریشه مضاعف و مقدار آن گزارش شود

■ در غیر این صورت ریشه های معادله محاسبه شده و نمایش داده شوند



تمرین ب

■ برنامه ای بنویسید که ساعات کارکرد کارمند و تعداد فرزندان وی را گرفته و حقوق آنها محاسبه نماید

■ حقوق خالص = ساعات کارکرد * ۲۵۰۰

■ به ازای هر فرزند و حداکثر تا سه فرزند، مبلغ ۱۵۰۰ تومان به حقوق اضافه میشود

■ اگر ساعات کارکرد از ۱۰۰ ساعت کمتر باشد به ازای ساعات باقیمانده تا ۱۰۰ برای هر ساعت مبلغ ۵۰۰ تومان از حقوق کسر میشود

■ مبلغ ۵٪ حقوق بعنوان مالیات و بیمه از حقوق کسر میشود.



ساختار انتخاب switch-case

- از این ساختار برای تصمیم گیری‌های چندگانه براساس مقادیر مختلف **یک عبارت**، استفاده میشود.
- بطور کلی در تمامی تصمیم گیری هایی که بیش از سه انتخاب وجود داشته باشد، بهتر است از ساختار **switch** استفاده شود.
- ساختار انتخاب **switch-case** مشابه **if-else** های متوالی میباشد ولی کد برنامه را خواناتر میکند.

نحوه استفاده از ساختار switch-case

switch (**<عبارت>**)

{

case **<مقدار ثابت ۱>**:
 <دستورات ۱>
 break;

case **<مقدار ثابت ۲>**:
 <دستورات ۲>
 break;

default:
 <دستورات n>

}

■ مقادیر نوشته شده در مقابل عبارت **case** باید **ثابت** باشند.

■ انتهای دستورات مربوط به هر **case** با کلمه کلیدی **break** مشخص میشود

■ در صورتی که حاصل ارزیابی عبارت در هیچ یک از حالات در نظر گرفته شده، موجود نباشد عبارت مربوط به بخش **default** اجرا میگردد.

نحوه استفاده از ساختار switch-case

switch (x)

{

case 1:

<دستورات ۱>

break;

case 7:

<دستورات ۲>

break;

case 9:

<دستورات ۳>

break;

case 15:

<دستورات ۴>

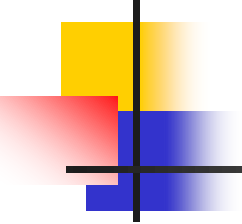
break;

default:

<دستورات ۵>

}

- فرض کنید متغیری با نام X وجود دارد که میخواهیم براساس مقادیر مختلف آن تصمیمات مختلفی بگیریم.
- اگر X برابر با یک بود، مجموعه دستورات ۱،
- اگر برابر با ۷ بود، مجموعه دستورات ۲،
- اگر برابر با ۹ بود، مجموعه دستورات ۳،
- اگر برابر با ۱۵ بود، مجموعه دستورات ۴
- و اگر با هیچکدام برابر نبود، مجموعه دستورات ۵ اجرا شوند.



نکات استفاده از ساختار switch

- ساختار switch میتواند فاقد بخش default باشد
- مقادیر موجود در case های switch نمیتوانند با هم مساوی باشند.
- توجه داشته باشید که ثوابت کاراکتری هنگام کامپایل به کد اسکی تبدیل میشوند
- اگر در انتهای دستورات مربوط به یک case، دستور break فراموش شود، دستورات مربوط به case بعدی نیز اجرا میشوند.



تفاوت‌های if-else و ساختار switch

- در ساختار **if** چند عبارت با هم مقایسه میشوند و شرطی که در **if** اول نوشته میشود محدودیتی در نوشتن **if** دوم ایجاد نمیکند ولی در **switch** حاصل یک عبارت مورد ارزیابی قرار میگیرد
- مقادیر مقایسه شونده در ساختار **if-else** میتوانند متغیر باشند ولی در ساختار **switch** مقادیر انتخاب باید ثابت‌هایی باشند که مقابل دستور **case** قرار میگیرند
- در ساختار **if-else** میتوان عبارت منطقی یا رابطه ای را مورد بررسی قرار داد ولی در ساختار **switch** فقط “مساوی بودن” مورد بررسی قرار میگیرد.

مثال:

- برنامه ای بنویسید که دو عملوند و یک عملگر را خوانده و با توجه به عملگر نتیجه را در خروجی نمایش دهد

```
int main()
{
    int num1 , num2 , result = 0;
    char op;
    printf("\nEnter num1: ");
    scanf( "%d" , &num1);
    printf("\nEnter num2: ");
    scanf( "%d" , &num2);
    printf("\nEnter operator(+,*,-,/): ");
    op = getche();
    switch(op)
    {
        case '+':
            result = num1 + num2;
            break;
        case '-':
            result = num1 - num2;
            break;
        case '*':
            result = num1 * num2;
            break;
```

```
        case '/':
            result = num1 / num2;
            break;
        default:
            printf("\n\noperator is undefined");
    }
    printf("\n\nresult is: %d",result);
    getch();
    return 0;
}
```

Enter num1: 5

Enter num2: 3

Enter operator(+,*,-,/): +

result is: 8