

(1)

تابع بازگشتی زیر را در نظر بگیرید و به سوال های زیر پاسخ دهید.

```
int f(int *n , int L )
{
    if(L==1)
        return n[L-1];

    else {
        int m=f(n , L-1);
        return ( ( n[L-1] < m)? (n[L-1]) : (m) ) ;
    }
}
```

a. بعد از اجرای قطعه کد زیر، چه عبارتی در خروجی چاپ خواهد شد؟ (با traceهای جداگانه)

```
int a[1] = { 10 }, b[4] = { 20,100,400,16 };
cout << f( a , 1) <<endl<< f(b,4 )<<endl;
```

b. این تابع چه عملیاتی را شبیه سازی می کند.(در یک جمله بیان کنید)

(a)

10

400

(b)

بزرگترین عضو ارایه را مشخص میکند.

(2)

خطا(های) موجود در برنامه زیر را اصلاح نمایید.(با ذکر دلیل)

```
#include<iostream>

using namespace std;

int main()
{
    int n,*ptr=&n;

    cin>>n>>ptr;

    *ptr=n-2;

    if (*ptr>=0) *n++;

    else

        cout<<*ptr <<" IS NEGATIVE\n";

    if (&n==ptr)

        cout<<"CORRECT POINTER";

    else

        ptr=&n;
```

```
    return 0;
}
```

اشاره گر پی تی از به ادرس ان اشاره دارد و نیازی به دریافت از کاربر ندارد.

```
;cin >> n
```

ان از نوع عدد صحیح است و برای یک واحد افزایش مقدارش نیاز به ستاره نیست.

```
;++n
```

(3)

برنامه ای بنویسید که درآمد ماهانه یک فروشگاه را در یک بازه زمانی مشخص از سال، دریافت نماید و میانگین درآمد ماهانه و کل درآمد آن فروشگاه را در آن بازه زمانی محاسبه و چاپ نماید. برنامه شما باید دارای بخش های زیر باشد:

برای ذخیره درآمدهای ماهیانه، از یک آرایه پویا استفاده نمایید.

- برنامه، تعداد اعضای آرایه را از کاربر دریافت می نماید.
- برنامه، از تابعی برای محاسبه میانگین درآمد ماهانه و از تابعی دیگر برای محاسبه کل درآمد استفاده می نماید.

یک نمونه از اجرای برنامه :

Please input the number of monthly sales to be input

4

Please input the sales for month 1

1290.89

Please input the sales for month 2

905.95

Please input the sales for month 3

1567.98

Please input the sales for month 4

994.83

The total sales is 4759.65

The average monthly sale is 1189.91

```
<include <iostream>
```

```
;using namespace std
```

```
;(double average(double *, int
```

```
;(double total_Sum(double *, int
```

```
0)int main
```

```
);
```

```
;int n
```

```
;(double *sales = new double[n
```

```
;":cout <<"Please Enter the number of monthly sales to be inopte
```

```
;cin >> n
```

```
(++for (size_t i = 0; i < n; i
```

```
}
```

```
;": " >> cout <<"please input sale for month " <<i + 1
```

```
;[cin >> sales[i
```

```
{
```

```
cout <<endl
```

```
" :the monthly average is" >>
```

```
;(average(sales, n >>
```

```
cout <<endl
```

```
" :the total sum is" >>
```

```
;(total_Sum(sales, n >>
```

```
;delete[] sales
```

```
;return 0
```

```
{
```

```
(double average(double *_sales_ptr, int _n
```

```
)
```

```
;double total_Av = 0
```

```
(++for (size_t i = 0; i < _n; i
```

```
}
```

```
;([total_Av += (_sales_ptr[i
```

```
{
```

```
;total_Av /= _n
```

```
;return total_Av
```

```
{
```

```
(double total_Sum(double *_sales, int _n
```

```
)
```

```
double total_S = 0;

(++for (size_t i = 0; i < _n; i

}

;[total_S += _sales[i

{

;return total_S

}
```

(4

تابعی بنویسید که یک ماتریس مربعی از اعداد صحیح را 90 درجه در جهت عقربه های ساعت بگرداند. ماتریس اولیه باید به عنوان ورودی تابع، در اختیار تابع قرار گیرد و ماتریس جدید باید به عنوان خروجی تابع (صریح، و نه ضمنی)، برگردانده شود. در شکل روبه رو، یک نمونه برای یک ماتریس 3*3 نشان داده شده است.

$$\begin{bmatrix} 11 & 22 & 33 \\ 44 & 55 & 66 \\ 77 & 88 & 99 \end{bmatrix} \Rightarrow \begin{bmatrix} 77 & 44 & 11 \\ 88 & 55 & 22 \\ 99 & 66 & 33 \end{bmatrix}$$

<--[if !vml][endif]--!>

```
<include <iostream>#

;using namespace std

define n 3#

([void returner(int _matrix[n][n], int _returned_Matrix[n][n]

)

(++for (size_t i = 0; i < n; i

}

(++for (size_t j = 0; j < n; j

}

;[returned_Matrix[j][i] = _matrix[j][_

{

{

(++for (size_t i = 0; i < n; i

}

;[int temp = _returned_Matrix[j][0

;[returned_Matrix[j][0] = _matrix[j][2_
```

{

5

```
0int main
```

Y

```
(++for (size_t i = 0; i < n; i
```

}

{

```
;(returner(matrix, returned_Matrix
```

```
(++for (size_t i = 0; i < n; i
```

}

{

5