Fall 2018 CS31 midterm 2

12' 1.

```
int dosomething (int a[], int n, int target){
    int begin = 0;
    int end = n-1;
    while (begin <= end){
        int mid = (begin + end)/2;
        if (a[mid]<target)
            end = mid -1;
        else if (a[mid] > target)
            begin = mid +1;
        else
            return mid;
    }
    return begin;
}
```

Elements in array are in strictly decreasing order.

```
int main(){
    int array[10];
    cout << dosomething (array, 10, Val);
}
```

Choices:

Function 1) returns the position of the largest element
2) returns the position of the largest element
3) returns number of elements less than Val
4) returns number of elements less or equal than Val
5) returns the position of element equal to Val
(6) returns number of elements more than Val
7) returns number of element more or equal to Val

A: (6)

15' 2.

```
const int X = 8 ; (the last digit of your UID)
int g (int k){
    return k+4;
}

void f (int n, int& d, int e, int c){
  switch (n){
      case 1:
          d += 3;
          e += 3;
          c += 3;
      break;
      case 2:
          cout << d << endl;
          cout << e << endl;
    }
}

int main(  ){
    int a = X;
    int b = X;
    int c = X;
   f(1, a, b, c);
   cout << g(c) << endl;
   f(2, a, b, c);
}
```

What's the output; _____

A: 12
11
8

11' 3.

cstring

```
const int X = 8; (the last digit of your UID)
void function ( char s[] ){
    int n= strlen( s ) -1;
    char ch;
    for ( ch = s[n]; n>1; n-- ){
        s[n] = s[n-1];
    }
    s[1] = ch;
}

int main( ){
    char word[10];
    switch(X){
        case 1: ...
        case 2: ...
        case 3: ...
        case 4: ...
        case 5: ...
        case 6: ...
        case 7: ...
        case 8: char word[10] = "SOUTH"; break;
        case 9: ...
        case 0: ...
    }
    function ( word );
}
```

what's word now?

A: SHOUT

10' 4. Write a function:

replace every element larger than target to the value of the target

return the number of elements not been replaced.

A: eg.

```
int function ( int a[ ], int n, int limit ){
    int count = 0;
    for ( int i=0; i<n; i++ ){
        if ( a[i] > limit )
            a[i] = limit;
        else
            count++;
    }
    return count;
}
```

15' 5. Write a function:

Tracey has 5 widgets.

a record of transactions. eg. "BSBBSSBBS" "SSSSSSSBB"

return the final amount left

set the reference parameter to the max amount ever in transaction

if string contains chars not 'B' or 'S' return -1;

if the total amount ever becomes negative return 0;

```
int main( ){
    str1 = "SBBSBBSSBSS"   str2 = "SSSSSSBB"
  assert ( analyze( str1, m) == 4 && m==7);
  assert ( analyze( str2, m) == 0)
  cout << "Correct!";
}
```

A: eg.

```
int analyze (string transactions, int& m){
    int amount = 5, max_amount = 5;
    for ( int i =0 ; i < transactions.size( ); i++){
       if (!( transactions[i] == 'B' || transactions[i] == 'S' ))
           return -1;
    }
    for (int i=0; i< transactions.size( ); i++){
       if ( transactions[i]== 'B')
           amount++;
       if ( transactions[i] =='S')
           amount--;
       if ( amount <0)
           retun 0;
       if ( amount > max_amount)
           max_amount =amount;
    }
    max = max_amount;
    return amount;
}
```