

Supreme-Batch-Debug-Exercise-C++ (Week-1)

NOTE: The code snippet given may be incomplete or have compile time, runtime or logical errors.

How to attempt Debugging Exercise?

1. Copy the code to your code editor (e.g. VS Code).
2. Add relevant header files like "#include <iostream>" etc.
3. Run the code.
4. You will notice the expected output is not printing at the console.
5. Apply your smart coder mind to Debug the code.
6. Warning: Only see the solution after you have tried enough.

1. Debug the code. Take input a character, print 1, if its a capital alphabet, print 0, if its a lowercase alphabet, else print -1.

```
#include<iostream>
using namespace std;

int main() {
    char c;
    cin>>c;
    if('a'<=c || c<= 'z'){
        cout<<0;
    }
    else if('A'<=c || c<= 'Z'){
        cout<<1;
    }
    else{
        cout<<-1;
    }
}
```

Handwritten notes on the right side of the code:

- A, B, C → 0
- A, B, C → 1
- 2, 3 → -1

2. Debug the code. It is trying to print the given pattern.

```
/*
Pattern
1
23
345
4567
*/
#include<iostream>
using namespace std;

int main(){
    int n; // cin >> n; Take input
    int i=1;
    while(i<n){ // loop must include the last row
        int j = i, count = 1;
        while(count<i){ // outer loop → row
            cout<<j;
            j = j + 1;
            count = count + 1;
        }
        cout<<"\n";
        i = i + 1;
    } // return;
}
```

Handwritten notes on the right side of the code:

- No of rows = 4
- outer loop → row
- inner loop → column

3. Debug the code. It is trying to print the given pattern.

```

/*
Pattern
N = 4
    *
   ***
  *****
 *****
*/
#include<iostream>
using namespace std;
int main(){
    int n;
    cin>>n;
    int i=1;
    while(i<=n){ → loop must run for the last row
        int j = i, count = 2*n+1, gaps=(n-i-1), k=1;
        while(k<gaps){ → (n-i)
            cout<<" ";
            k = k + 1;
        }
        int m = 1, ch = count - 2*gaps;
        while(m<ch){
            cout<<"*";
            m = m + 1;
        }
        k = 1;
        while(k<gaps){
            cout<<" "; ←
            k = k + 1;
        }
        cout<<"\n";
        i = i + 1;
    } → return 0
}

```

4. Debug the code. It is trying to print the given pattern.

```

/*
Pattern
N = 4
1
22
333
4444
*/
#include <iostream>
using namespace std;

int main(){
    int i,j,n;
    cin>>n;
    for(i=1;i<n;i++){
        for(j=1;j<n;j++){
            cout<<i;
        }
        cout<<endl;
    }
}

```

```

#include <iostream>
using namespace std;

int main() {
    int i, j, n;
    cin >> n;

    // FIX 1: Loop should run from 1 up to and including n.
    for (i = 1; i <= n; i++) {
        // FIX 2: Inner loop should run 'i' times.
        for (j = 1; j <= i; j++) {
            cout << i;
        }
        cout << endl;
    }
    return 0;
}

```

5. Debug the code. It is trying to print the given pattern.

```

/*
Pattern
N = 4
1
21
321
4321
*/
#include<iostream>
using namespace std;

int main(){
    int i,j,n;
    cin>>n;
    for(i=1;i<n;i--){
        int p;
        for(j=1;j<n;j++){
            cout<<p;
            p--;
        }
        cout<<endl;
    } → return 0
}

```

int main () {
 (i=1;j<n;j++)
 p=1
 for (j=1;j<=i;j++)
}

```

#include <iostream>
using namespace std;

int main() {
    int i, j, n;
    cin >> n;

    // FIX 1: The outer loop must increment from i=1 up to n.
    for (i = 1; i <= n; i++) {
        // FIX 2: Initialize 'p' with the current row number.
        int p = i;

        // FIX 3: The inner loop must run 'i' times for row 'i'.
        for (j = 1; j <= i; j++) {
            cout << p;
            p--; // Decrement p to print the countdown sequence.
        }
        cout << endl;
    }
    return 0;
}

```

6. Debug the code. It is trying to print the given pattern.

```
/*
Pattern
N = 5
E
DE
CDE
BCDE
ABCDE
*/
```

```
#include<iostream>
using namespace std;

int main() {
    int i,j,n;;
    cin>>n;
    for(i=1;i<=n;i++){
        char p='A'+i-1;
        for(j=1;j<=i;j++){
            cout<<p;
            p++;
        }
        cout<<endl;
    }
    return 0;
}
```

1 2 3 4 5
A B C D E

E
DE
CDE
BCDE
ABCDE

for row i=1 & n=5, 'A'+5-1 = 'A'+4 → A
for row i=5 & n=5, 'A'+5-5 = 'A'+0 → A

7. Debug the code. It is trying to print the given pattern.

```
/*
Pattern
N = 4
1
232
34543
4567654
*/
```

```
#include<iostream>
using namespace std;
int main(){
    int no;
    cin>>no;
    int i=1;
    int n = 2*no;
    while(i<=(no)){
        int gaps = n-2*i+1,k=1;
        int j = i;
        while(k<=gaps/2){
            cout<<" ";
            k = k + 1;
        }
        int ch = n - gaps + 1, z = (ch+1)/2;
        while(z>1){
            cout<<j;
            j = j - 1;
            z = z + 1;
        }
        j = j - 1;
        z = (ch-1)/2;
        while(z>=1){
            j = j - 1;
            cout<<j;
            z = z - 1;
        }
        k = 1;
        while(k<=gaps/2){
            cout<<" ";
            k = k + 1;
        }
        cout<<"\n";
        i = i + 1;
    }
}
```

1 → Infinite loop

2 → Incorrect printing logic

3 → Overly complicated

```
#include <iostream>
using namespace std;
```

```
int main() {
    int n;
    cin >> n;
    int i = 1;
```

```
while (i <= n) {
    // 1. Print leading spaces
    int space = n - i;
    while (space > 0) {
        cout << " ";
        space--;
    }
}
```

// 2. Print the first half (increasing numbers)

```
int j = i;
int count = 1;
while (count <= i) {
    cout << j;
    j++;
    count++;
}
```

// 3. Print the second half (decreasing numbers)

```
j = j - 2; // Reset j to the correct starting number
count = 1;
```

```
while (count < i) {
    cout << j;
    j--;
    count++;
}
```

```
cout << endl;
i++;
```

```
}
```

```
return 0;
```

$n-i$
for rows (i=1) \rightarrow
row 4 (i=4) \rightarrow

$(n-i)$ spaces

8. Debug the code. It is trying to print the given pattern.

part ① half + Middle

$5 \rightarrow \frac{5+1}{2} = 3$

$i = 20$
 $3-i = 3-1 = 2$ $\rightarrow *$
 $3-i = 3-2 = 1$ $\rightarrow **$
 $3-i = 3-3 = 0$ ~~***~~

$\star \rightarrow 1, 3, 5$

part -2 [bottom]

- print space $=$
- print star \star

```

/*
Pattern
N = 5
*/
void printPatt(int n){
    int i=1;
    while(i<(n)){
        int gaps = n-2*i,k=1;
        if(i>(n)/2){
            int no = (n+1)/2;
            gaps = 2*(i%no);
        }
        while(k<=gaps/2){
            cout<<" ";
            k = k + 1;
        }
        int ch = n - 1 - gaps;
        while(ch>=1){
            cout<<"*";
            ch = ch - 1;
        }
        k = 1;
        while(k<=gaps/2){
            cout<<" ";
            k = k + 1;
        }
        cout<<"\n";
        i = i + 1;
    }
}

```

Solutions

```

void printPatt(int n) {
    // This pattern works best with an odd number for 'n'.
    // Part 1: Print the top half of the diamond (including the middle row)
    int top_rows = (n + 1) / 2;
    for (int i = 1; i <= top_rows; i++) {
        // Print leading spaces
        for (int j = 1; j <= top_rows - i; j++) {
            cout << " ";
        }
        // Print stars
        for (int k = 1; k <= 2 * i - 1; k++) {
            cout << "**";
        }
        cout << endl;
    }
    // Part 2: Print the bottom half of the diamond
    int bottom_rows = n / 2;
    for (int i = 1; i <= bottom_rows; i++) {
        // Print leading spaces
        for (int j = 1; j <= i; j++) {
            cout << " ";
        }
        // Print stars
        for (int k = 1; k <= n - 2 * i; k++) {
            cout << "**";
        }
        cout << endl;
    }
}

```

int size
cout << "Enter an odd number: "
cin >> size;
if (size % 2 != 0) {
 printPatt(size);
} else {
 cout << "Please enter an
odd number: " << endl;
}
return 0;