- STRICTLY follow the input and output format. Don’t use filler statements like printf(“Enter the number: “);  
- Never use getch(). It is included in the non-standard header file conio.h.  
- If the output for each test case is to be printed on different line, use printf(“\n”); or cout<<endl; after each test case.  
- Use ‘int main’ instead of 'void main'. return 0 at the end of each program. If the main program returns 0, means it was executed successfully.  
- For a sample problem statement: There are t test cases. For each test case, you have to input space separated numbers n and m and output the greater on a new line.   
<stdio.h>  
 int main()  
 {  
int t, m, n;  
scanf(“%d”, &t);  
while(t--)  
{  
scanf(“%d %d”, &m, &n);  
if(m>n)   
printf(“%d”, m);  
else   
printf(“%d”, n);  
printf(“\n”);  
}  
return 0;  
}  
- If the input size is more than 10^5, declare the arrays globally. Local arrays more than this size will throw runtime error.  
- ALWAYS remember the range of int and long long int. If the numbers are less than 10^9, int can be used safely. Range of long long is 10^18. Declare variables accordingly.  
- For input/output, scanf/printf are faster than cin/cout. If you face TLE, try to optimize the code first. If it is mentioned to use faster I/O, use accordingly.  
- While scanning characters or strings in C, put a space before %c or %s to avoid taking the enter key as an input character. e.g “ %c” and “ %s”.

What is debugging?  
Debugging is a methodical process of finding errors in your code and removing those errors each line at a time.

Tips for debugging  
- Check if you have & before all your scanf variables.  
- Check if you have all larger arrays declared globally.  
- Check if you have used ‘==’ for comparison and not the ‘=’ assignment operator.  
- Print variable values at specific steps to check the correctness of your code upto that step.