DEBUGGING AND TESTING

Debugging is the process of locating and fixing or bypassing bugs (errors) in computer program code or the engineering of a hardware device. To *debug*a program or hardware device is to start with a problem, isolate the source of the problem, and then fix it. A user of a program that does not know how to fix the problem may learn enough about the problem to be able to avoid it until it is permanently fixed. When someone says they've debugged a program or "worked the bugs out" of a program, they imply that they fixed it so that the bugs no longer exist.

Debugging is a necessary process in almost any new software or hardware development process, whether a commercial product or an enterprise or personal application program. For complex products, debugging is done as the result of the unit test for the smallest unit of a system, again at component test when parts are brought together, again at system test when the product is used with other existing products, and again during customer beta test, when users try the product out in a real world situation. Because most computer programs and many programmed hardware devices contain thousands of lines of code, almost any new product is likely to contain a few bugs. Invariably, the bugs in the functions that get most use are found and fixed first. An early version of a program that has lots of bugs is referred to as "buggy."
Debugging tools (called *debugger*s) help identify coding errors at various development stages. Some programming language packages include a facility for checking the code for errors as it is being written.

- **Debugging:**

Debugging is the process of detecting and correcting the syntax errors in a program.
This consists of two stages:

1. Systematic desk checking
2. Translator system checking

1. Desk Checking

When a program is completed, a thorough desk check will eliminate many needless errors. This check may include are view by a second person who is an expert in computer programming.

2. Translator Checking

'C' system has certain rules which must be followed, such as rules for naming variable and for using punctuations. If rules are not followed or if the instructions are not coded properly, the translator system will detect these errors during the translation process.

- **Program Efficiency:**

A program utilize two major resources of a computer system, namely, computer memory and computer time. These resources are very costly and therefore programs should be designed to use them to the minimum possible extent. A program which requires less memory and takes minimum time to execute is called an efficient program. However, the prime concern of a programmer should be the correctness of the program rather than its efficiency, there is no use of designing an efficient program if it cannot produce correct result. The general rule, therefore first produce a correct program and then optimize its resources requirements.

Memory:

Space requirements of a program become too critical when a limited memory is available for use. In a multiprocessing environment, the system can take more programs at a time and process them simultaneously if the space needed for each program is less.

1. Select an algorithm which needs fewer steps. Remember it should not be at the cost of accuracy.
2. Minimize the number of variable requires.
3. When a variable is not required any more in the program, reuse it rather than selecting another variable.
4. Minimize the use of string variables.
5. Do not store heading lines in memory; use them in printf() statement directly as string constant.
6. When we use array variables declare their sizes exactly in declaration statement.
7. Do not break large expressions into small bits using new variables.
8. Do not read anything which can be calculated inside the program.
9. Wherever possible use self replacement techniques.

Execution Time:

Time is costliest resource and a programmer who could come up with ideas and tricks to make a program run faster are considered a good programmer. Saving in time becomes more important when a program is to be used quite often.

- **Testing:**

Testing is a process of identifying defects, where a defect is any different between actual and expected result.

Error

Any problem in program is known as Error.

Describe the types of error

Ans:

There are following categories of errors:
1. Syntax errors
2. Logical errors
3. Data errors

1. Syntax Errors

Syntax Errors in computer programs that typically involve incorrect punctuation, incorrect word sequence, undefined terms, or misuse of terms.

These errors are automatically detected, and pointed out by language processor.

For example:

```
int   X(10)
pintf(“Syntax Error”);
clrser();
int x=y+/2;
```

All have syntax errors. All such errors should be corrected before running the program.

2. Logical Errors

Errors like taking a wrong path, failure to consider a particular condition, incorrect order of evaluation of expressions, incorrect order of evaluation of statements, etc., belong to this category. These errors are primarily due to poor understanding of the problem, incorrect translation of algorithm into program and abuse of brackets and operators.

For example:

Writing if A < B ….. in place of IF A > B

Writing if X = Y ….. when X and Y are real numbers.

Are valid but might produce incorrect results. Sometimes such errors might create infinite loops.

3. Data Errors

Data errors are the most common errors. This may be due to wrong typing of constants, mismatching of variable list and data order, and failure to assume the range of values a variable might take in the program.

For example:

asking the computer to divide a quantity by zero, or to compute the square root of a negative number are due to failure to anticipate the ranges of data.

Short Questions:

- ✓ What is the goal of program testing?
- ✓ What is Syntax error? Give example.
- ✓ Define testing.
- ✓ Give the use of debugging.

Long Questions:

- ✓ What do you mean by program efficiency? Discuss any one source of program efficiency.
- ✓ Write a short note on debugging.
- ✓ What is debugging? Explain in detail.