**Answer any 5(1 Score Each)**

1. 2 2. + 3. goto 4. translation 5. rhombus 6. 4

**Answer any 9 from 7-18. (2 Score Each)**

7. The stage where programming errors are discovered and corrected

8. Time consuming, laborious, difficulty to modify, no standards any of these two

9. 32B is invalid. starts with digit (1 Score)
float is a keyword(1 Score)

10. digit is identifier 54.6 is literal( ½ Score Each)
switch is keyword % is operator

11. // for single line comment and /*... */ for multiline commenting (1+1=2 Score)

12 ++ increment by one
-- decrement by one (1+1=2 Score)

13. i. 0(false) ii. 1(true) (1+1=2 Score)

14. pi can't be assigned new value (1Score)
because it is declared constant. (1 Score)

15. If no match is found with any one of the value provided in case statements, then the statements in default gets executed . (2 Score)

16. Accessing each element of the array once
Arranging elements of the array in some logical order (1+1=2 Score)

17. i. M (1 Score) ii. WELCOM (1Score)

18. i. \0(1Score) ii. Last element of any string in c++ is null character (1Score)

**Answer any 9 from 19-30(3 Scores Each)**

19. i. Rules of the programming language gets violated. Incorrect punctuation, incorrect word sequence, undefined term, or illegal use of terms or constructs.(1 Score)
ii. Logical error occurs due to improper planning of program's logic. (1 Score)
iii. Interruption in program execution due to inappropriate data encountered during operation in execution. (1 Score)

20. Flowchart to print 1 to 10 (3 Score)

21. *Entry Controlled*: Condition check before body execution. Body may never executed. Suitable to skip body part logic.
*Exit Controlled:* Condition check after body execution. Body executed at least once. Suitable to execute body part logic at least once.
( 1½+1½=3 Score)

22. Logical &&, Logical || logical ! and explanation (1+1+1=3 Score)

23. *break :* Used with switch and loops, Brings the program control outside the switch or loop by skipping the rest of the statements within the block. Program control goes out of the loop even though the test expression is true.
*continue :* Used only with loops. Brings the program control to the beginning of the loop skipping the rest of the statements within the block.
Program control goes out of the loop only when the test expression becomes false( 1½+1½=3 Score)

24. switch (n) (3 Score)
```
{
case 1: cout<<"One"; break;
case 0: cout<<"Zero";break;
default:
        cout<<"Not a binary number";
}
```

25. *While:* Entry controlled loop. Initialisation before loop definition. No guarantee to execute the body of the loop at least once
*Do-While:* Exit controlled loop. Initialisation before loop definition. Will execute the body of the loop atleast once even though the condition is false
( 1½+1½=3 Score)

26. if(n1>n2) big=n1;
else big=n2; ( 3 Score)

27.*Linear Search:* The elements need not be in any order. Takes more time for the process. May need to visit all the elements. Suitable when the array is small. ( 1½ Score)
*Binary Search:* The elements need to be in any order. Takes very less time for the process. All the elements are never visited. Suitable when the array is large. ( 1½ Score)

28: ( 3 Score)
```
#include<iostream>
using namespace std;
int main()
{int a[5];
cout<<"Enter 5 elements";
cin>>a[0]>> a[1]>> a[2]>> a[3]>> a[4];
cout<< a[4] <<a[3]<< a[2]<< a[1]<< a[0];
return 0; }
```

29. i. 1 ii. 8 iii. 3 (1+1+1=3 Score)

30. A sorting technique by repeatedly stepping through lists that need to be sorted, comparing each pair of adjacent items and swapping them, if they are in the wrong order. This passing procedure is repeated until no swaps are required, indicating that the list is sorted. Gets its name because, larger element bubbles towards the top of the list.(3 Score)

**Answer any 2 from 31,32,33 (5 Scores Each)**

31. Problem identification- preparing algorithms and flowcharts-coding the program using programming language- translation- debugging-execution and testing-documentation

32 .i. 1 ii.11 iii. 10 iv. 10 v. 1 (5 Score)

33.i. int m[3][4]; (1 Score)
ii. cout<<m[1][2]; (1 Score)
iii. cout<<m[2][3]; (1 Score)
iv. (2 Score)
```
for(i=0;i<3;i++)
    for(j=0;j<4;j++)
        cout<<m[i][j];
```

**Prepared by: Roy John,**
**St. Aloysius H S S,**
**Elthuruth**