Question 1

Which special symbol allowed in a variable name? *Mark 1*

a)  

b)  

c)  

d)  

**Answer:** d)

**Explanation:** As per the Syntax of the language. Refer Slides

Question 2

Which of the following are unary operators in C? *Mark 1*

a)  

b)  

c)  

d)  

**Answer:** b) d)

**Explanation:** As per the Syntax of the language. Refer Slides

Question 3

Which of the following declarations are correct? *Mark 1*

a) struct mystruct {int a;};

b) struct {int a;}

c) struct mystruct {int a;}

d) struct mystruct int a;

**Answer:** a)

**Explanation:** As per the Syntax of the language. Refer Slides.
**Question 4**

What will the function `Sum` return? *Mark 1*

```c
void sum(int x, int y) {
    x++; y++;
    return (y);
}
```

a) The incremented value of y  
b) The incremented value of y; the value of x is incremented but not returned  
c) Compilation Error: return value type does not match the function type  
d) Does not incremented value of y

**Answer:** c)

**Explanation:** The return type of the function is void, hence an integer value cannot be returned.

**Question 5**

What value will be printed for `data.c`? *Marks 2*

```c
#include<stdio.h>  
#include <string.h>  

int main() {  
    union Data {  
        int i;  
        unsigned char c;  
    } data;  
    data.c = 'C';  
    data.i = 89;  
    printf( "%c\n", data.c);  
    return 0;  
}
```

a) C  
b) Y: ASCII 89  
c) G  
d) C89

**Answer:** b)

**Explanation:** When `%c` is used for printing an integer value, conversion to the equivalent ASCII...
**Question 6**

What is the output of the above program? *Marks 2*

```c
#include <stdio.h>
void foo(int[]);
int main()
{
    int myarray[4] = {1, 2, 3, 0};
    foo(myarray);
    printf("%d ", myarray[0]);
}
void foo(int p[4])
{
    int k = 34;
    p = &k;
    printf("%d ", p[0]);
}
```

a) 1 2  
b) 1 3  
c) Will always output 1  
d) 34 1  

**Answer:** d)  
**Explanation:** The base pointer of the array is used to point to an integer 34. In main, the array is accessed directly to print the 1st element.

**Question 7**

What is the output of the following program? *Marks 2*

```c
#include <stdio.h>
#define func(x, y) x / y + x
int main()
{
    int i = -6, j = 3;
    printf("%d
",func(i + j, 3));
    return 0;
}
```

a) divide by zero error  
b) -4  
c) -8  
d) 3  

**Answer:** c)  
**Explanation:** \(x/y+x\) replaced by \(i + j/3 + i + j\) i.e \((-6 + 3/3 -6 +3) = (-6 + 1 -6 +3) = -8\)
Question 8

What will be the output of the following program? *Marks 2*

```c
#include <stdio.h>
int sum(int a, int b, int c) {
    return a + b + c / 2;
}
void main() {
    int (*function_pointer)(int, int, int);
    function_pointer = sum;
    printf("%d", function_pointer(2, 3, 4));
}
```

a) Compilation Error: Error in function call
b) 7
c) 4.5
d) 5.5

**Answer:** b)

**Explanation:** *function_pointer* is a pointer defined for any function with 3 integer parameters and integer return type. It points to function *sum* and returns the result of the sum.

Question 9

Fill in the blank to concatenate strings *str1* and *str2* to form *str3*? *Marks 2*

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

int main(void) {
    string str1 = "I ";
    string str2 = "Travel";

    string str3 = ______________________;
    cout << str3;
    return 0;
}
```

Output: I Travel

a) *str1*+*str2*
b) *strcat*(*str1*,*str2*);
c) *strcat*(*strcpy*(*str3*,*str1*),*str2*);
d) *str1*.append(*str2*)

**Answer:** a) *str1*+*str2* and d) *str1*.append(*str2*)

**Explanation:** *str1* and *str 2* are two string type variables, operations possible for concatenation are *str1*+*str2* (String is a stl, hence has + operator overloaded) and *str1*.append(*str2*) to append strings.
Question 10

What will be the output of the following program? Marks 2

```cpp
#include <iostream>
#include <algorithm>
using namespace std;
bool srt (int i, int j) {
    return (i < j);
}
int main() {
    int data[] = {52, 76, 19, 5, 10, 100, 56, 98, 17};
    sort(data + 1, data + 4, srt);
    for (int i = 0; i < 9; i++)
        cout << data[i] << " ";
    return 0;
}
```
a) 52 5 19 76 10 100 56 98 17  
b) 52 76 19 10 5 100 56 98 17  
c) 76 5 10 19 76 100 56 98 17  
d) 76 5 10 19 76

Answer: a)  
Explanation: The whole array is not passed for sorting, only from index 1 (data + 1, i.e. 0 + 1) to index 4 (data + 4, i.e. 0 + 4), i.e. 3 elements, 76, 19, 5

Question 11

What will be the output of the following program? Marks 2

```cpp
#include <iostream>
#include <string.h>
#include <stack>
using namespace std;
int main() {
    char str[19] = "Programming";
    stack<char> s;
    for (int i = 0; i < strlen(str); i++)
        s.push(str[i]);
    for (int i = 0; i < strlen(str) - 1; i++) {
        cout << s.top();
        s.pop();
    }
    return 0;
}
```
a) rogramming  
b) ogramming  
c) gnimmargor
d) gnigormmar

**Answer:** c)

**Explanation:** When programming pushed to stack, the element on the top is g (gnimmargorp), which is displayed and then popped. Continues till length of str - 1, hence p not printed at the end.

**Question 12**

Fill up the blanks for A# and B# below: Marks 2

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

int main() {
    cout << "Enter the no. of elements: ";
    int count, j, sum=0;
    cin >> count;
    --------------------- A# // Declare with Default size
    --------------------- B# // Change the size to the required amount
    for(int i = 0; i < arr.size(); i++) {
        arr[i] = i;
        sum += arr[i];
    }
    cout << "Array Sum: " << sum << endl;
    return 0;
}
```

a) A#: `vector <int> arr(count);`
   B#: `arr.resize(count);`

b) A#: `vector <int> arr(count);`
   B#: `arr.size(count);`

c) A#: `vector <int> arr;`
   B#: `arr.size(count);`

d) A#: `vector <int> arr;`
   B#: `arr.resize(count);`

**Answer:** d)

**Explanation:** As per syntax, using resize operator