2022 Academic Challenge Computer Science Sectional Exam

Question 1:
Which of the following is the operating system kernel not responsible for:

a) Memory allocation
b) Implementing the GUI *
c) Resource management
d) Input/Output device control
e) Fault tolerance

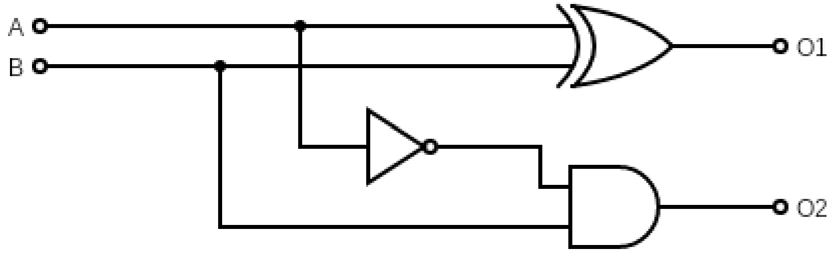
Question 2:
Which of the following is not a data serialization format:

a) XML
b) JSON
c) YAML
d) XDR
e) ALGOL*

Question 3:
Why do network cables like those classed as Cat 6 ethernet cables use twisted-pair wiring?

a) To make the wires easier to bundle
b) To reduce resistance and attenuation in the wires
c) To reduce interference and crosstalk *
d) Because they carry binary data
e) Because the signal is polarization modulated

Question 4:
What operation does the depicted circuit implement?



a) Half adder
b) Full adder
c) Adder-subtractor
d) Half subtractor *
e) Full subtractor

Question 5

What is output to the console when the following program is run?

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

int main()
{
    double x = 1.5;
    int y = 4, z = 1;
    for(int i = 0; i < y; ++i) {
        z = x*z;
    }
    cout << "z = " << z << endl;
    return 0;
}
```

a) z = 1 *

b) z = 5.0625

c) z = 7

d) z = 16

e) Compile-Time Error

Question 6

What is the decimal value of adding the hexadecimal numbers 0x8FB and 0x178?

a) 2675 *

b) 1673

c) 81688

d) 2477

e) None of the above

Question 7:

Which of the following are equivalent to $\overline{D \cdot (C \cdot D + B \cdot C)} + \overline{B} \cdot \overline{B}$

a) *B + C + D*

b) 1

c) 0

d) $\overline{B} + \overline{C} + \overline{D}$ *

e) $\overline{B} + \overline{C} \cdot \overline{D}$

Question 8:

Which of the following is true of double precision floating point (binary64) representations:

a) The significand has 64 bits of precision

b) The significand has 32 bits of precision

c) The representation for 0 is unsigned

d) There is no representation for NaNs

e) The absolute precision depends on the floating point value *

Question 9:
Which of the following is described as having nodes connected to an arbitrary number of neighbors such that there is a possible traversal from any node to another:

a) Star network
b) Fully connected network
c) Mesh network *
d) Tree network
e) Star network

Question 10:
Which of the following would not typically be included on system on a chip

a) Processor cores
b) Metal-oxide-semiconductor memory
c) Field-programmable gate arrays *
d) Random access memory
e) Interfaces

Question 11: What is the output when the following program is run?

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

int main()
{

    int a[] = {12, 7, 10, 10, 12, 7};
    int tmp = 0;

    for(int i = 0; i < 6; ++i) {
        cout << (tmp ^= a[i]) << " ";
    }

    return 0;
}
```

a) 12 15 15 15 15 15
b) 12 11 1 11 7 0*
c) 0 0 0 0 0 0
d) 12 84 840 8400 100800 705600
e) 35831808 -2147483648 -2147483648 -2147483648 -2147483648 (due to integer overflow)

For questions 12-14, refer to the following program:

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

class Shape
{
protected:
    int w, h;
public:
    Shape() : w(1), h(1) {}
    Shape(int a, int b) : w(a), h(b) {}

    double area()
    {
        return -1;
    }

    void print()
    {
        cout << "Area = " << area() << endl;
    }

};

class Triangle : public Shape
{
public:
    Triangle(int a, int b) : Shape(a, b) {}
    double area()
    {
        return w*h/2;
    }
};

class Rectangle : public Shape
{
public:
    Rectangle(int a, int b) : Shape(a, b) {}
    double area()
    {
        return w*h;
    }
};

int main()
{
    Rectangle r(5, 10);
    Triangle t(5, 10);

    Shape *s_r = &r;
    Shape *s_t = &t;

    s_r->print();
    s_t->print();

    return 0;
}
```

Question 12: What is printed to the console when the program is run?

a) Area = -1 *
   Area = -1

b) Area = 25

Area = 50

c) Area = 50
   Area = 25
d) Compile-time error
e) Run-time error

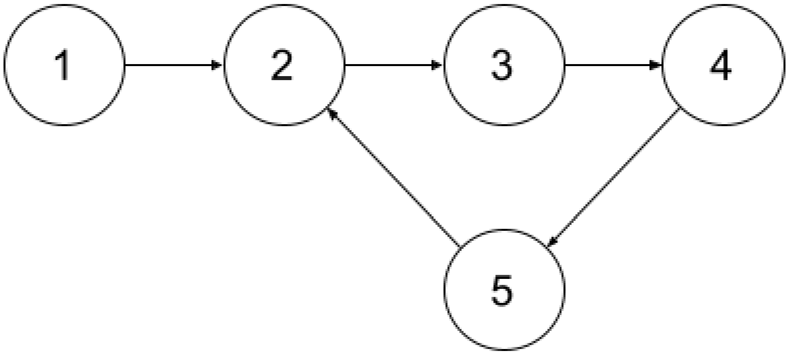
Question 13: If line 12 is modified to be `virtual double area().` What is printed to the console when the program is run?

a) Area = -1
   Area = -1
b) Area = 25
   Area = 50
c) Area = 50 *
   Area = 25
d) Compile-time error
e) Run-time error

Question 14: The section of code described in Question 13 is an example of what concept?

a) Multiple Inheritance
b) Polymorphism *
c) Virtual Machines
d) Recursion
e) Imaginary Overloading

For Questions 15 - 17, consider the following figure and code:



```
01  #include <iostream>
02  using namespace std;
03
```

```
struct Node {
    int val;
    Node* next;
    Node(int n) : val(n), next(NULL) {}
    Node(int n, Node* p) : val(n), next(p) {}
};

bool fun(Node *head)
{
    if(head==NULL||head->next==NULL) return false;
    Node* a = head->next->next;
    Node* b = head->next;
    while(a!=NULL && a->next!=NULL) {
        if(a==b||a->next==b) return true;
        a = a->next->next;
        b = b->next;
    }
    return false;
}

void print(Node *tmp)
{
    int i = 0;
    while(i < 6) {
        cout << tmp->val << " ";
        tmp = tmp->next;
        ++i;
    }
    cout << endl;
}

int main()
{
    /*
      See Question 17
    */

    /*
      See Question 18
    */

    return 0;
}
```

Question 15: Select the data structure which best describes the figure above.

a) Binary tree
b) Directed graph *
c) Recursive queue
d) Cyclic stack
e) Lollipop array

Question 16: Consider the three options below. When included in the `main()` function shown above, select which of these can represent the structure from Question 15 and will print `1 2 3 4 5 2` to the console when the code is compiled and run.

(I)

```
Node* n1 = new Node(1);
Node* n2 = new Node(2);
Node* n3 = new Node(3);
Node* n4 = new Node(4);
Node* n5 = new Node(5);

n1->next = n2;
n2->next = n3;
n3->next = n4;
n4->next = n5;
n5->next = n2;

print(n1);
```

(II)

```
Node* n5 = new Node(5);
Node* n4 = new Node(4,n5);
Node* n3 = new Node(3,n4);
Node* n2 = new Node(2,n3);
Node* n1 = new Node(1,n2);
n5->next = n2;

print(n1);
```

(III)

```
Node n1(1), n2(2), n3(3), n4(4), n5(5);
n1.next = &n2;
n2.next = &n3;
n3.next = &n4;
n4.next = &n5;
n5.next = &n2;

print(&n1);
```

a) II only
b) III only
c) I & II
d) I & III
e) I, II, & III *

Question 17: Consider making a call to function `fun`. Describe what `fun` does, and what will be the return value.

a) Determines if the data structure contains a palindrome i.e. reads the same forwards and backwards. The return value is `false`.
b) Determines if the data structure contains an even number of elements. The return value is `false`.
c) Determines if the data structure contains an even number of elements. The return value is `true`.

d) Determines if the data structure contains a cycle. The return value is `true`. *

e) Reverses the ordering of the connections between nodes. Returns `true` when the reversal is complete.

For questions 20 and 21 consider the following program:

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

class X
{
private:
    stack<int> a, b;

    void consolidate()
    {
        if(b.empty()) {
            while (!a.empty()) {
                b.push(a.top());
                a.pop();
            }
        }
    }

public:
    void push(int x)
    {
        a.push(x);
    }

    int pop()
    {
        consolidate();
        int top = b.top();
        b.pop();
        return top;
    }

    int& peak()
    {
        return b.top();
    }

    bool empty()
    {
        return a.empty() && b.empty();
    }
};

int main()
{
    int a[5] = {4, 10, 14, 18,  8};
    X myStructure;
    for(int i = 0; i < 5; ++i) {
        myStructure.push(a[i]);
    }

    myStructure.pop();
    myStructure.peak() += 5;

    while(!myStructure.empty()) {
        cout << myStructure.pop() << " ";
    }
    cout << endl;
    return 0;
}
```

Question 18:

What is the data structure implemented by class X:

a) Stack
b) Queue *
c) Doubly linked stack
d) Double-ended Queue
e) Linked lists

Question 19:

What is the output of the program

a) 8 18 14 10
b) 8 18 14 15
c) 10 14 18 8
d) 15 14 18 8 *
e) 9 10 14 18 8

Question 20: Consider the code snippet below:

```
enum month {Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec};

void season(int m)
{
    switch(m) {
    case(m >=  3 && m <  6):
        cout << "spring" << endl;
    case(m >=  6 && m <  9):
        cout << "summer" << endl;
    case(m >=  9 && m < 11):
        cout << "autumn" << endl;
    case(m >= 11 || m <  3):
        cout << "winter" << endl;
    }
}
```

What would be the result of making the function call `season(Nov)` ?

a) `Autumn` is printed to the console
b) `Winter` is printed to the console
c) Run-time error because the enumeration definition is invalid
d) Compile-time error because the enumeration definition is invalid

e) Compile-time error because the cases are not constructed with valid constant expressions *

Question 21:
Which of the following is not true of a static data member of a class?

a) The member is initialized before any object of the class is created
b) The member is deleted when the last object of the class is destroyed*
c) Only one copy of the member is created
d) Each object of the class can access the the member
e) The member can be accessed without any class object being created

For questions 22 and 23 consider the following program:

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

int k(int m, int n)
{
    if (m == 0) return n + 1;
    if (n == 0) return k(m - 1, 1);
    return k(m - 1, k(m, n - 1));
}

int main()
{
    cout << "k(1, 2)=" << k(1, 2) << endl;
    return 0;
}
```

Question 22
What is the output of the program?

a) k(1, 2) = 1
b) k(1, 2) = 2
c) k(1, 2) = 3
d) k(1, 2) = 4 *
e) k(1, 2) = 5

Question 23
Which of the following is false about function k

a) The function k is an example of a recursive function
b) The function evaluations grows faster than linearly in m
c) The function evaluations grows faster than linearly in n
d) The function can be converted to an iterative solution *
e) The function is a total computable function

For questions 24 and 35 consider the following program:

```
#include <iostream>
#include <vector>
```

```cpp
using namespace std;

int fun(vector<int> v)
{
    int a = 0, b = 0;

    for (int i = 0; i < v.size(); i++) {
        b += v[i];
        b = max(b, 0);
        a = max(a, b);
    }
    return a;
}

int main()
{
    vector<int> v{ 2, -1, 3, -2, 5,  2, -1, -3, 1 };
    cout << fun(v) << endl;

    return 0;
}
```

Question 24: What is output to console when the program is run?

a) 7
b) 8
c) 9 *
d) 10
e) 0

Question 25: The code above is an implementation of an algorithm which determines the maximum sum of any contiguous sub-array of v. In order to return the intended quantity, what criteria must the input array satisfy?

a) At least one element must be non-negative *
b) There must be at least as many positive elements as there are negative
c) The sum of all the elements must be non-negative
d) The first element of the input array must be positive
e) The input array can be any set of `int`.

For Questions 26-30, refer to the following code:

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

int funA(int n)
{
    if(n == 1 || n == 2) return 1;
    return funA(n-1) + funA(n-2);
}

int funB(int n)
```

```
{
    int x = 1, y = 1;
    for(int i = 2; i < n; ++i) {
        int z = x + y;
        x = y;
        y = z;
    }
    return y;
}

int funC(int n, vector<int>& v)
{
    if(n == 1 || n == 2) v[n-1] = 1;
    if(v[n-1] > 0) return v[n-1];
    return (v[n-1] = funC(n-1, v) + funC(n-2, v));
}

int main()
{
    int n = 6;
    vector<int> v(n,-1);
    cout << "A = " << funA(n) << ", B = " << funB(n) <<
        ", C = " << funC(n,v) << endl;
    return 0;
}
```

Question 26:
What is output to the console after running the program?

a) A = 8, B = 6, C = 8
b) A = 8, B = 8, C = 8 *
c) A = 13, B = 8, C = 8
d) A = 8, B = 8, C = -1
e) A = 8, B = 13, C = 8

Question 27:
Which of the functions shown above will produce the same return value for an arbitrary positive integer n?

a) funA, funB, and funC *
b) funA and funB
c) funB and funC
d) funA and funC
e) None of the above

Question 28:
Which of the following best describes the time complexity of funA, funB, and funC respectively?

a) O(N!), O(1), O(N)
b) O(N), O(N), O(N)
c) O(N), O(1), O(N)*
d) O(2^N), O(1), O(N)
e) O(2^N), O(N), O(N)

Question 29:

Which of the following best describes the memory complexity of funA, funB, and funC respectively?

a) O(N), O(N), O(N)
b) O(N), O(1), O(N)
c) O(2^N), O(1), O(N)*
d) O(2^N), O(N), O(N)
e) O(2^N), O(1), O(1)

Question 30:

What is the programming technique employed in funC called?

a) Template Programming
b) Memoization *
c) Memory Leak
d) Iteration
e) Type Casting