INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

Department of Computer Science and Engineering

Kanwal Rekhi Building

CS101 Makeup Exam with Answers

Sunday, 9 November 2014

[Please DO NOT turn this page till start of examination is announced]

Important:

1. There are 15 questions. You are expected to answer all questions.

2. Please remember to write your answer on both answer sheets supplied, immediately after attempting each question.

3. Please do not try speculative guess work to avoid penalty of negative marks

Q1. Which of the following are <u>VALID</u> datatypes in C++

A. number B. float C. int D. long long int

E. char F. short float G. double int

Answer: B, C, D, E

Q2. What is the result of execution of the following C++ program if the user enters "thequickbrownfox" when prompted to enter 'myString'?

```
#include <iostream>
using namespace std;
int main()
{
    char myString[12];
    cout << “Enter the string: “;
    cin >> myString;
    cout << myString;
    return 0;
}
```

Select from the folowing which are <u>VALID</u>:

A. The program gives compilation error.

B. The program will compile successfully and print the string “thequickbrownfox” without any error messages

C. The program will compile and print the string “thequickbr” without any error messages

D. The program will compile successfully but will likely crash

E) None of these

Answer: D

Q3. Convert the hexadecimal number 'A2B' into decimal, binary, and octal.

A. **Decimal**: 2603, **Binary**: 101000101011, **Octal**: 05053

B. **Decimal**: 2602, **Binary**: 101010101010, **Octal**: 05054

C. **Decimal**: 2603, **Binary**: 101000111001, **Octal**: 05052

D. **Decimal**: 2604, **Binary**: 101010100011, **Octal**: 05053

Answer: A

Q4. Consider the program given below that takes two numbers of float type as input from the user. The two numbers are divided, and their result is stored in two different variables 'num1' and 'num2' of float and double type.

```
#include<iostream>
using namespace std;
int main() {
   float num1;
   double num2;
   float num3, num4;
   cin >> num3 >> num4;
   num1 = num3 / num4;
   num2 = num3 / num4;
   if(num1 == num2)
     cout<<"Both the numbers are equal\n";
   else
     cout<<"Both the numbers are not equal\n";
   return 0;
}
```

Assume that num3 is always greater than num4 and num4 is not zero. Select from the folowing which are <u>VALID</u>:

A. The program will always output '**Both the numbers are equal**', irrespective of the input entered by the user for **num3** and **num4**, where **num3 > num4**.

B. The program will always output '**Both the numbers are not equal**', irrespective of the input entered by the user for **num3** and **num4**, where **num3 > num4**

C. The program may or may not output '**Both the numbers are equal**' depending on the input the input entered by the user for **num3** and **num4**, where **num3 > num4**

D. None of these

Answer: C

Q5. A Chinese credit card company named 'Gooda Banka' offers some credit points on a certain amount of purchase made by the customer.

Their scheme is as follows :

If the purchase made is from Rs. 200/- to Rs. 500/-, then they offer 2 credit points on every 100 Rupees spent

If the purchase made is from Rs. 501/- to Rs. 750/-, then they offer 3 credit points on every 90 Rrupees spent

If the purchase made is from Rs. 751/- to Rs. 999/-, then they offer 4 credit points on every 80 Rupees spent

If the purchase made is from Rs. 1,000/- to Rs. 1,500/-, then they offer 6 credit points on every 65 Rupees spent.

Any other purchase amount should be considered **INVALID**

To attract more customers, they have added one more criterion. i.e. if the customer is associated with the bank for more than two years, then credit points earned will be increased by 25 points on every purchase irrespective of the amount. The following program tries to compute the credit points earned by a customer from 'Gooda Banka'.

```
#include<iostream>
using namespace std;
int main() {
   float creditPurchase=0, creditPoints=0;
   int years=4;
   cout <<"Enter the amount spent using credit card"<<endl;
   cin >> creditPurchase;
   if(creditPurchase < 200 || creditPurchase > 1500)
     cout << "Invalid credit purchase\n";
   else if(creditPurchase == 200 || creditPurchase <= 500)
     creditPoints += ((creditPurchase / 100) * 2);
   else if(creditPurchase == 501 || creditPurchase <= 750 )
     creditPoints += ((creditPurchase / 90) * 3);
   else if(creditPurchase == 751 || creditPurchase <= 999)
     creditPoints += ((creditPurchase / 80) * 4);
   else
     creditPoints += ((creditPurchase / 65) * 6);

   if (years > 2)
     creditPoints += 25;

   cout << "The credit points earned on purchase of " <<
            creditPurchase << " is " << creditPoints << endl;
   return 0;
}
```

Select from the folowing which are VALID. Assume credit purchase is always positive.

A. The program computes the credit points correctly, only if the credit purchase is between 200 and 750

B. The program computes the credit points correctly, only if the number of years of the customer associated with the bank is greater than two years

C. Irrespective of the credit purchase amount entered by the user, the program computes the credit points correctly if the number of years of the customer associated with the banks is less than two years

D. None of these

Answer: B

Q6. Consider the following program

```
#include<iostream>
using namespace std;
int main(){
   int num1=4;
   int counter=0;
   for(counter=0; counter<=4; counter+=2) {
      switch(counter) {
         case 1:
            num1++;
         break;
         case 2:
            num1+=2;
         case 4:
            num1%=2;
            counter=-1;
            continue;
         default:
            --num1;
            continue;
      }
   }
   return 0;
}
```

Select from the folowing which are <u>INVALID</u>:

A. The **default** case is executed twice, whereas, **Case 1** and **Case 2** are executed exactly once

B. **Case 1** is executed exactly once, whereas **Case 2** is executed exactly twice

C. **Case 4** is executed exactly twice, i.e. once immediately after executing **Case 2** and other, immediately after the counter is incremented by 2 in the for loop

D. After the end of the for loop, if we add a statement '**cout << counter - num1;**' in the above code, then the value displayed will be 4

Answer: B, C

Q7. Consider the following program

```
#include<iostream>
using namespace std;
int main(){
  int a, b;
  do{
    cout << "Enter a: ";
    cin >> a;
    cout << "Enter b: ";
    cin >> b;
  } while(a < 10 || b > 10);
  return 0;
}
```

Select from the folowing which are <u>INVALID</u>:

A. The program will always terminate for all values of input '**a**' greater than 9, irrespective of the value of input '**b**'

B. The program will always terminate for all values of input '**b**' lesser than 11, irrespective of the value of input '**a**'

C. For all values of input '**a**' greater than 9 and input '**b**' lesser than 11, the program is guaranteed to terminate

D. The program will always terminate for all positive values of input '**a**' and '**b**'.

E. None of these

Answers: A, B, D

Q8. Consider the following program

```
#include<iostream>
using namespace std;
int main(){
  int a, b;
  cout << "Enter a: ";
  cin >> a;
  cout << "Enter b: ";
  cin >> b;
  while(a < 20 && b > 20){
      cout << "Enter a: ";
      cin >> a;
      cout << "Enter b: ";
      cin >> b;
      if(a >= 20 || b <= 20) continue;
  }
  return 0;
}
```

Select from the folowing which are <u>VALID</u>:

A. The program will always terminate for all values of input '**a**' greater than 19, irrespective of the value of input '**b**'

B. The program will always terminate for some combination of input '**a**' and input '**b**'

C. For all values of input '**a**' lesser than 20 and input '**b**' greater than 20, the program is guaranteed to terminate

D. The program will never terminate for any positive values of input '**a**' and '**b**'.

E. None of these

Answers: A, B

Q9. Consider the program given below.

```
#include<iostream>
using namespace std;
int main() {
   int a, b, c, num1, num2;
   cout << "Enter two numbers \n";
   cin>>num1>>num2;
   if(num1 > num2) {
      a = num1;
      b = num2;
   }
   else {
      a = num2;
      b = num1;
   }
   while(St1) {
      St2
      a = b;
      b = c;
   }
   return 0;
}
```

Select the appropriate choice(s) from the following:

A. If we replace **`St1`** with '**`a!=0`**', **`St2`** with '**`c = b % a;`**', then the program will always compute the GCD of numbers '**num1**' and '**num2**' and store in variable '**b**'.

B. If we replace **`St1`** with '**`b!=0`**', and **`St2`** with '**`c = a % b;`**', then the program will always compute the GCD of numbers '**num1**' and '**num2**' and store in variable '**a**'.

C. If we replace **`St1`** with '**`b!=0`**', and **`St2`** with '**`c = a % b;`**', then the program does not necessarily compute the GCD of numbers '**num1**' and '**num2**' and store in variable '**a**'.

D. If we replace **`St1`** with '**`a!=0`**', and **`St2`** with '**`c = b % a;`**', then the program will always compute the GCD of numbers '**num1**' and '**num2**' and store in variable '**a'**.

Answer: B

Q10. Consider the following program.

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

int main(){
  int n=10;
  int A[n][n];
  int j,k;
  j=0;
  for(int i=0; i<n*n;i++){
    k=i%n;
    cin >> A[j][k];
    if(k == n-1)
       j++;
  }
  return 0;
}
```

Which of the following can be said about the above program? Select the ones which are appropriate:

A. Some of the elements of the 2-dimensional array '**A**' will contain values(garbage values) other than what the user enters

B. The program will initialize all the elements of a 2-dimensional array '**A**'

C. Only the first row of the 2-dimensional array '**A**' will get initialized

D. Only the first column of the 2-dimensional array '**A**' will get initialized

E. None of these

Answers: B

Q11. Consider the following program. Assume 10 integer values are supplied as input.

```
#include<iostream>
using namespace std;
int main(){
  int n=10;
  int A[n], B[n];
  int k=0;
  for(int i=0;i<n;i++)
    cin >> A[i];
  bool found=false;
  for(int i=0; i<n-1;i++){ //Begin first for loop
    found=false;
    for(int j=i+1;j<n;j++){ // Begin second for loop
      if((A[i]-A[j])==0){ // Begin first If
        for(int temp=k-1; temp>=0; temp--){// Begin third for loop
          if(A[i]==B[temp]){
            found=true;
            break;
          }
        } // End of the third for loop
        if(found==false){
          B[k]=A[i];
          k++;
        }
      } // End of the first if
    } // End of the second for loop
  } // End of the first loop
  return 0;
}
```

Which of the following can be said about the above program? Select the ones which are appropriate:

A. All duplicate elements of array '**A**' are stored in array '**B**' with as many duplicates as is present in array '**A**'

B. The program successfully finds all duplicate elements in array '**A**' and stores only one copy of the duplicates in the array '**B**'

C. The program may not find duplicates in array '**A**' for some input configuration

D. None of these

Answers: B

Q12. We want to write a C++ program that computes the integral part of the n^{th} root of an integer 'm', where both 'm' and 'n' are positive integers. Thus, if n = 2 and m = 7, then our function should compute the integral part of $7^{1/2}$ i.e. 2. Similarly, if n = 3 and m = 30, the function should compute the integral part of $30^{1/3}$ i.e. 3.

A student has written the C++ function shown below for this purpose:

```
int ComputeIntNthRoot(int m, int n){
  if(m==1) return 1;
  else if(n==1)return m;
  else{
    int result, i,j;
    for(i =2; C1 ; i++){
      result = 1;
      for(j=1 ; j <= n; j++){
        result*=i;
      }
      if(C2) return (i-1);
    }
  }

}
```

Which of the following boolean conditions C1 and C2 will cause the program to correctly compute the integral part of the n^{th} root of m for all values of m and n greater than or equal to 1?

A. C1: i <= m/2+1 C2: result > m

B. C1: i <= n C2: result > m

C. C1: i <= n/2 + 1 C2: result > n

D. C1: i <= m C2: result > n

E. None of these

Answers: A

Q13. In the following program, how many times is the function 'Recursive' called, including the call from the main function? Also, what will be the output of this program?

```
#include <iostream>
using namespace std;
int Recursive(int n)
{
  if (n == 0) return 1;
  if (n == 1) return 2;
  return (Recursive(n - 1) * Recursive(n - 2));
}
int main()
{
  int value=Recursive(5);
  cout << "Result: " << value <<endl;
  return 0;
}
```

Select the appropriate choice(s) from the following

A. The recursive function will be called 15 times and the following output is produced: Result: 32

B. The recursive function will be called 14 times and the following output is produced: Result: 32

C. The recursive function will be called 13 times and the following output is produced: Result: 28

D. The recursive function will be called 14 times and the following output is produced: Result: 18

E. None of these

Answer: A

Q14. We want to find if a string is a subsequence of another string. A subsequence is a sequence that can be derived from another sequence by deleting some elements without changing the order of the remaining elements. For example, the sequence "one" is a subsequence of "oranges".

Consider the following recursive function 'isSubSeq' to check if the character string in the array 'string1' is a subsequence of another character string in the array 'str2'. Assume that both the arrays 'string1' and 'string2' store a '\0'-terminated string. The function returns 'true' if the string in array 'string1' is a subsequence of the string in array 'string2', else it returns 'false'. In the function, variable 'i' is the length of string 'string1' and variable 'j' is the length of striing 'string2'.

Note that the code given below has two unspecified expression Expr1 and Expr2. Assume that the compiler directives '#include<iostream>' and 'using namespace std;' are present at the begining of the program file.

```
bool isSubSeq(char string1[], char string2[], int i, int j)
{
    if (i == 0) return true;
    if (j == 0) return false;
    if (string1[i-1] == string2[j-1])
        return Expr1;
    return Expr2;
}
```

Which of the following choices for `Expr1` and `Expr2` will result in the program doing exactly as the problem specification requires?

A. **`Expr1`** is '**`isSubSeq(string1, string2, i-1, j-1)`**'
`Expr2` is '**`isSubSeq(string1, string2, i, j-1)`**'

B. **`Expr1`** is '**`isSubSeq(string1, string2, i, j-1)`**'
`Expr2` is '**`isSubSeq(string1, string2, i, j)`**'

C. **`Expr1`** is '**`isSubSeq(string1, string2, i+1, j+1)`**'
`Expr2` is '**`isSubSeq(string1, string2, i, j)`**'

D. None of these

Answers: A

Q15. We want to count the number of words in a character string. The words can be separated by the following characters: space ' ', tab character '\t', new line '\n', or a combination of these. For example, the string "two oranges make one apple\n four bananas\t and five grapes" has 10 words.

Consider the following function 'numWords' which counts the number of words present in the character string 'str'. Assume that the array 'str' stores a '\0'-terminated string.

Note that the code given below has two unspecified assignments Assign1 and Assign2, and one unspecified condition Cond. Assume that the compiler directives '#include<iostream>' and 'using namespace std;' are present at the begining of the program file.

```
int numWords(char str[]){
    int notSeparator = 0;
    int wordCount = 0;
    int i = 0;
    while (str[i++]!='\0')
    {
        if (str[i] == ' ' || str[i] == '\n' || str[i] == '\t')
            Assign1;
        else if (Cond)
        {
            Assign2;
            ++wordCount;
        }
    }
    return wordCount;
}
```

Which of the following choices for **`Assign1`**, **`Cond`**, and **`Assign2`** will result in the program doing exactly as the problem specification requires?

A. **`Assign1`** is '**`notSeparator = 1`**',
`Cond` is '**`notSeparator == 1`**',
`Assign2` is '**`notSeparator = 0`**'

B. **`Assign1`** is '**`notSeparator = 0`**',
`Cond` is '**`notSeparator == 0`**',
`Assign2` is '**`notSeparator = 1`**'

C. **`Assign1`** is '**`notSeparator = 1`**',
`Cond` is '**`notSeparator == 0`**',
`Assign2` is '**`notSeparator = 1`**'

D. None of these

Answers: B

Best Wishes for a Good Score