To help us know more about you and your abilities we would like to hear some of your thoughts on the topics below along with your technical answers to the following coding and T-SQL questions.

Please take a few moments to answer the questions below to the best of your ability and once completed send your answers back via email.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name:** | Naing Min Thu | **Experience:** | Above 3 years |
| **Latest Employer:** | ACE Data Systems Co, Ltd. | **Latest Position:** | Software Engineer |
| **Availability** | <please enter> | **GitHub Portfolio URL:** | https://github.com/NaingMinThu/calculation\_project.git |

**Coding**

1. Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".  
     
   The program is to be written as a C# console application.

Optional: Extra marks will be given for applications written using TDD approach and showing the test methods.

1. Create a HTML page showing a list of 5 items from which the user can choose one at a time. When the user clicks an item an alert box will be displayed showing the content of the item that was clicked. The text of the item which is clicked will turn bold. When a new item is clicked the format from the first item will be removed.

When the user hovers over an item, the item will be highlighted.

The list should be centered on the page both horizontally and vertically.

Your CSS and JavaScript should be cross-browser compatible and at minimum work in the latest versions of Internet Explorer, Chrome and Firefox.

1. Programmatically find the least number of hops needed to get from Id=1 to Id=15 using the grid created by the SQL below. For example, to get from Id=1 to Id=4 will require 2 hops - (1,3) followed by (3,4).

CREATE TABLE #TESTHOPS ([FromId] INT, [ToId] INT)

INSERT INTO #TESTHOPS VALUES   
(1,2), (1,3), (3,4), (5,10), (5,6), (7,5), (1,7), (2,8), (8,9), (9,11), (9,10), (10,12), (10,14), (12,13), (14,15)

Please write your answer in the space below.

**Answer:**

CREATE TABLE #TESTHOPS ([FromId] INT, [ToId] INT)

INSERT INTO #TESTHOPS VALUES

(1,2), (1,3), (3,4), (5,10), (5,6), (7,5), (1,7), (2,8), (8,9), (9,11),

(9,10), (10,12), (10,14), (12,13), (14,15)

declare @startID int=1, @endid int=11

Create table #Tmp1(

RowID int identity(1,1),

FromID int,

ToID int

)

Declare @min int=1, @max int= (select COUNT(\*) from #TESTHOPS)

while @min<= @max

BEGIN

declare @count\_tmp int=(select COUNT(\*) from #Tmp1)

IF ((select top 1 FromID from #Tmp1 order by RowID desc) =@startID)

BEGIN

set @min = @max + 1

END

ELSE

BEGIN

IF(@count\_tmp = 0)

BEGIN

Insert into #Tmp1(FromID,ToID)

select top 1 \* from #TESTHOPS where ToId=@endid order by FromId desc

END

ELSE

BEGIN

Insert into #Tmp1(FromID,ToID)

select top 1 tp.FromID,tp.ToID from #TESTHOPS tp

inner join (select top 1 FromID from #Tmp1 where RowID = @min -1 order by RowID desc)as tmp on tmp.FromID = tp.ToId

order by FromID desc

END

set @min +=1

END

END

select \* from #Tmp1 order by RowID desc

drop table #Tmp1

drop table #TESTHOPS

**T-SQL**

1. Insert 100 rows into the following table:

CREATE TABLE #TEST1 (TEST\_ID INT IDENTITY(1,1))

**Answer:**

CREATE TABLE #TEST1 (TEST\_ID INT IDENTITY(1,1))

SET IDENTITY\_INSERT #TEST1 ON;

DECLARE @numRows int,@i int

SET @numRows = 100

SET @i=1

WHILE @i<= @numRows

BEGIN

INSERT #TEST1(TEST\_ID) SELECT @i

SET @i=@i+1

END

SET IDENTITY\_INSERT #TEST1 OFF;

SELECT \* FROM #TEST1

DROP TABLE #TEST1

1. Consider a table with an [Id] column as well as BIT columns named [A] to [E] and some initial values:

CREATE TABLE #TEST2 ([Id] INT, [A] BIT, [B] BIT, [C] BIT, [D] BIT, [E] BIT)  
INSERT INTO #TEST2 ([Id], [A], [C], [E]) VALUES (1, 'true', 'false', 'true')  
INSERT INTO #TEST2 ([Id], [A], [B], [C]) VALUES (2, 'true', 'true', 'true')  
INSERT INTO #TEST2 ([Id], [C], [D], [E]) VALUES (1, 'false', 'false', 'true')

Create a procedure which will accept parameters ([Id] INT, [Column] CHAR (1)) and then invert the value in table #TEST2 for row matched by [Id] and column named [Column]. No change is made if existing value is NULL.

**Answer:**

Create PROCEDURE tsp\_InvertValue

@id int,

@column char(1)

AS

BEGIN

CREATE TABLE #TEST2 ([Id] INT, [A] BIT, [B] BIT, [C] BIT, [D] BIT, [E] BIT)

INSERT INTO #TEST2 ([Id], [A], [C], [E]) VALUES (1, 'true', 'false', 'true')

INSERT INTO #TEST2 ([Id], [A], [B], [C]) VALUES (2, 'true', 'true', 'true')

INSERT INTO #TEST2 ([Id], [C], [D], [E]) VALUES (1, 'false', 'false', 'true')

Declare @Query nvarchar(500)

set @Query = 'Update #TEST2 set '+ @column +'= case when '+@column

+' IS NULL then '+@column +' else case when '+ @column +

' = 1 then 0 else 1 end

end from #TEST2 where Id= '+cast(@id as nvarchar(5)) +''

exec(@Query)

select \* from #TEST2

drop table #TEST2

END

GO

1. Write SELECT query to list **latest** record for each [Id]:

CREATE TABLE #TEST3 ([Id] VARCHAR(10), [Data] VARCHAR(250), [Date] SMALLDATETIME)

INSERT INTO #TEST3 VALUES   
('A', 'abc', '2016-10-01'),  
('A', 'def', '2015-05-20'),  
('B', 'xyz', '2014-05-20'),  
('B', 'uvw', '2016-10-01'),  
('B', 'rst', '2015-10-01')

**Answer:**

CREATE TABLE #TEST3 ([Id] VARCHAR(10), [Data] VARCHAR(250), [Date] SMALLDATETIME)

INSERT INTO #TEST3 VALUES

('A', 'abc', '2016-10-01'),

('A', 'def', '2015-05-20'),

('B', 'xyz', '2014-05-20'),

('B', 'uvw', '2016-10-01'),

('B', 'rst', '2015-10-01')

select tmp1.Id as ID,tmp.MaxDate as [Date],tmp1.Data from #TEST3 tmp1

inner join ( select Id, max([Date]) as MaxDate

from #TEST3

group by Id)tmp on tmp.Id = tmp1.Id and tmp.MaxDate = tmp1.[Date]

drop table #TEST3

**General Questions**

1. How would you measure the quality of source code?
2. In what circumstances would code require refactoring?
3. What do you feel is the most promising new technology in web / software development and why?