bitnine

# 2022 Apache AGE Internship Program

## [ Coding Test ]

PURE

Progressiv
e Universa
l Respectf
ul Eccentr
ic

OCTOBER 2022

# Coding Test Questions Available

- ## Backend development –2 Questions

**Developing Apache AGE requires a lot of logical thinking that backend development often requires.**

- ## Frontend development –2 Questions

**Graph representation of data (using nodes and edges) is an essential feature of Graph databases and it requires a lot of heavy JavaScript programming.**

- ## Database Driver (Interface) –1 Question

**Database driver is a piece of software that allows applications to connect to and interact with the database system. Understanding how database drivers work is important for core programming required to develop Graph database.**

**Please be advised that the coding test will NOT play any significant role in determining roles and tasks of the interns for Apache AGE and AGE Viewer projects.**

**Please choose the question(s) that you are confident in explaining your answer / codes.**
**You can take any number of questions.**
**BUT please mind answer both Q1 and Q2 if you choose questions about C language**

# [ Question No. 1 ]

**Please write this in C**

**Node is defined as follows :**

```
typedef struct Node
{
    TypeTag type;
} Node;

typedef enum TypeTag {
...
}
```

Please submit
- Source code
-README text file explaining development environment and how to use / run the function.

Using this structure, please write a function that returns fibonacci sequence based on the following arithmetic operations (+, -, *, /) and conditions. Data type is int (int only) and the fibonacci function should be implemented using Dynamic Programming(DP).

```
main(){
Node *add = (*makeFunc(ADD))(10, 6);  N
ode *mul = (*makeFunc(MUL))(5, 4);  Node
*sub = (*makeFunc(SUB))(add, mul);
Node *fibo = (*makeFunc(SUB))(abs(sub), NULL); // Get n-th from fibonacci numbers.

calc(add);
calc(mul);
calc(sub);
calc(fibo)
}

Return
 - add : 16
 - mul : 20
 - sub : -4
 - fibo : 2
```

## [ Question No. 2 ]

**Please write this in C**

**There are two ways that you can tackle this and you could use either one**

Please submit
- Source code
-README text file explaining development environment and how to use / run the function.

Way 1)

Please implement the following piecewise recurrence relation in the 3 generally known, but distinctly different ways.
Please explain the differences (advantages, disadvantages) between the 3.
Please only code in C.

Way 2)

Please implement the following piecewise recurrence relation in the way that you feel is the best.
Please explain why it is the best.
Please only code in C.

**The problem**

**F(n) = F(n-3) + F(n -2) where F(0) = 0, F(1) = 1, and F(2) = 2.**
**Assume that n will be less than or equal to the maximum integer value and non-negative. You only need to write the function(s).**

**[ Question No. 3 ]**

Use any map api to draw the Los Angeles' subway network.

Please submit
- HTML and JavaScript files
- README text file explaining development environment and how to view the map

**[ Question No. 4 ]**

Using the Apace EChart's Graph (https://echarts.apache.org/examples/en/editor.html?c=graph-label-overlap), please draw the Los Angeles' subway network.

# Coding Test – Driver (Interface) [Question No. 5]

The open-source PostgreSQL drivers are available for various programming languages, including C++, Java and Node.js.
Using the language that you are most confident in, please modify the PostgreSQL driver source to return outputs in JSON string format (like the following).
Please submit the source code, driver file and a instruction for using the driver.

```
{
   "status_code" : 200
   ,"data" : [
      {"user_id":1,"name":"John","age":28}
      ,{"user_id":2,"name":"Tom", "age":29," phone":"1-800-123-1234"}
      ,{"user_id":3,"name":"Jenny", "age":34}
   ]
}
```

Please submit
- Source code
- Driver file
-README text file explaining development environment and how to use the driver

```
# Postgresql table
CREATE TABLE IF NOT EXISTS public.user_table
(
user_id numeric(10,0) NOT NULL,
name character varying(50) COLLATE pg_catalog."default" NOT NULL,
age numeric(3,0) NOT NULL,
phone character varying(20) COLLATE pg_catalog."default",
CONSTRAINT user_table_pkey PRIMARY KEY (user_id)
);
INSERT INTO public.user_table (user_id, name, age, phone) VALUES (3, 'Jenny', 34, NULL);
INSERT INTO public.user_table (user_id, name, age, phone) VALUES (2, 'Tom', 29, '1-800-123-1234');
INSERT INTO public.user_table (user_id, name, age, phone) VALUES (1, 'John', 28, NULL);
```

bitnine

# Thank You

**PURE** Progressive•Universal•Respectful•Eccentric