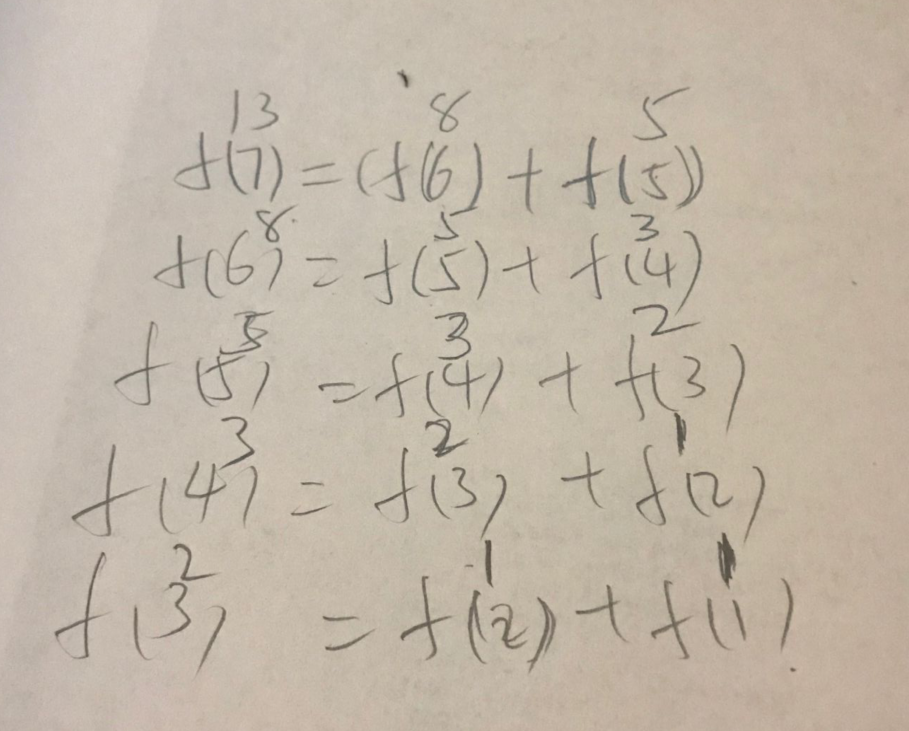
4/11/



2. give me a function int f(int n). which will return the nth element in the array:      
C:\Users\Yu\Documents\Tencent Files\651412330\Image\C2C\5M233LW4UR~1}55ZO_E[M{Q.png  
  
hint.  F(n) = F(n-1) + F(n-2)

1. do a recursion function int sum( int n )  
if input is n, you reutrn 1+2+... \_+ n

3/30

class design practice. write classes in c++

Create classes for Truck, Car and SUV. Your design must meet the following requirments:

1. Those three wil have shared attibutes: 1. horse power(int). 2. seat number(int). 3. MPG(int). 4. Make(stirng). 5. Model(stirng). 6. Vehicle Type(enum, google how to use enum in c++) These shared attributes must be put in the parent class Vehicle. Make sure you understand the meaning of the word in attribute. If you don't know, you can search google, i.e., what is the MPG of a vehicle?

2. These attributes CANNOT be accessed directly from outside of the class (trying to use the attritbutes will cause a build error). Must use member function GetAttributeName or SetAttributeName.

3. Each child class has it's own attributes:. For truck, it has its own member Towing Capacity (float). For car, it has its own member Car Type (it should be an enum of Coupe and Sedan, if you don't know how to use enum, you can google). For SUV, it has its own member AWD (bool, check the meaning of AWD if you don't now). And their own get and set. These members are also not allowed to be accessed from outside, need to use Get and Set to access.

4. Set the Vehicle Type (enum) in the constructor.

5. If an object of Truck, Car and SUV, is pointed by a Vehicle pointer, you should be able to call the member function "GetVehicleType" of Vehicle to get the type of the object. It should be an enum of Truck, Car or SUV. Here there is NO need to use virtual function.

6. If an object of Truck, Car and SUV, is pointed by a Vehicle pointer, you should be able to call the member function "GetMonthlyCost" of vehicle to get different monthly cost for the truck, car and suv. For truck, the monthly cost is MPG \* Tow Capacity / 100; for car, the montly cost is MPG \* 30 for sedan, MPG \* 35 for coupe; for SUV the monthly cost is MPG \* 40 for not AWD, MPG \* 45 for AWD. This one use virtual function.

7. You can build it without any problems.

8. You can run the following main function and get correct results. And you can get build failure after you uncomment line "std::cout << v->mpg << std::endl; "

int main()

{

Car car1, car2;

Truck truck1, truck2;

SUV suv1, suv2;

Vehicle \*v = nullptr;

car1.SetMPG(30);

car1.SetCarType(CAR\_TYPE\_SEDAN); // this will work after you add the enum

car2.SetMPG(30);

car2.SetCarType(CAR\_TYPE\_COUPE); // this will work after you add the enum

truck1.SetMPG(20);

truck1.SetTowingCapacity(6000);

truck2.SetMPG(20);

truck2.SetTowingCapacity(5000);

suv1.SetMPG(25);

suv1.SetAWD(true);

suv2.SetMPG(25);

suv2.SetAWD(false);

v = &car1;

// this line SHOULD cause a BUILD FAILE.

// After you cause the build fail, you can comment it out.

// if you cannot cause build fail. Your code is wrong.

// std::cout << v->mpg << std::endl;

// This will work after you add the enum

if (v->GetVehicleType() == VEHICLE\_TYPE\_CAR) {

std::cout << "test 1 passed\n";

}

else {

std::cout << "test 1 failed\n";

}

v = &car1;

if (v->GetMonthlyCost() == 900) {

std::cout << "test 2 passed\n";

}

else {

std::cout << "test 2 failed\n";

}

v = &truck2;

if (v->GetMonthlyCost() == 1000) {

std::cout << "test 3 passed\n";

}

else {

std::cout << "test 3 failed\n";

}

}

3/29

// check if a positive integer n can be represented by sum of some consecutive integers.

// if n = 100.

// then output 100 = 9 + 10 + 11 + 12 + 13 + 14 + 15 + 16

// 100 = 18 + 19 + 20 + 21 + 22

// Two error in the code.

#include <stdio.h>

void fun(int n)

{

int j, b, c, m, flag = 0;

for (b = 1; b <= n / 2; b++) {

n = m; // error m = n

c = b;

while (m != 0 && m >= c) {

m = m - c;

c++;

}

if (m != 0) { // error m == 0

printf("%d = ", n);

for (j = b; j < c - 1; j++)

printf("%d + ", j);

printf("%d\n", j);

flag = 1;

}

}

if (flag == 0) {

printf("Couldn't find.");

}

}

int main()

{

int n;

printf("input: ");

scanf\_s("%d", &n);

fun(n);

}

3/27

Hint. Don't forget the all of the strings has a hidden char = 0 or '\0' and the end.

1. Suppose the function "fun" has two inputs string s ( char \*) and char c, the function will search in s to find if there is a char having same value as c. if found, insert c again after this char.

If input s = "bbaacda" c = 'a' . then result s = "bbaaaacdaa"

try to find what to put in the blank [1] [2] and [3]

[1]: ‘\0’

[2]: ‘0’

[3]: ‘c’

any memory that allocated to you is random. if you don't do initialize. anything is random.

1. 1MB can hold how many integers?

1MB= 1024 KB

1KB = 1024 byte

1024\*1024/4= 1048576/4=262144

2. What is high level language and low-level language?

Human languages: Chinese, English

High level language: C++, java

Low level language: assembly is the normal language

Machine language: binary.

3. Multithreading, Mutex & Semaphore

Multithreading: share memory in the process (data, code, heap), each threading has own stack,

register,

mutex: have one lock

semaphore: have more than one lock.

Mutex and semaphore work between threads or process.   
4. Virtual Memory (Paging)

Each process has 4G memory, have many partitions(4K). Each partition address on virtual memory and physical memory saved to the page table.

5. What is heap, stack in memory?

Heap: dynamic memory, shared memory.

Stack: static memory, for each thread.

6. How to allocate array? How to allocate linked list?

Array: Continues

Linked list: could link to different location.

7. How to delete the entire linked list?

Delete the value in the linked list and unlink it.

8. What is "buffer"?

Like input, waiting for something.

Wrong. Buffer is like a block of contiguous memory to hold data. Input is just an specific example. Anything like:

char array[500];

int arry1[1000];

are buffers.

9. What is Object Oriented Programming?

Everything is class. Constructor

10. What is inheritance?

Son has all the features of the father.

Call son’s construction, also will call the father’s constructor.

11. What is polymorphism? (Function overloading and overriding?)

Some type of things. Dog: jingba or heibei .

You need to be more specific. like different between function overloading and overriding. Will test you later.

12. What is Standard Template Library (STL) in C++?

Download other people’s library, when you use it , just include the library and use their functions.

Wrong. STL are standard libraries like vector, queue, list, stack what are provided by C++. it comes with template. when you declare, you do:

std::vector<int> a;

std::queue<float> b;

different template holds differnt type of data.

1. Which of the following is a correct comment?

/\* \*/

Top of Form

**2.Which of the following is true?**

~~if(1)?~~

~~if(66)~~

i~~f(.1)?~~

~~if( -1)?~~

All of the above\*

Top of Form

**3.Evaluate! (1 && ! (0 || 1)).**

True\*

~~False?~~

~~Unevaluatable?~~

**4.Which is not a loop structure?**

For

Do while

While

Repeat Until\*

Top of Form

**5.When does the code block following while(x<100) execute?**

When x is less than one hundred?\*

~~When x is greater than one hundred?~~

~~When x is equal to one hundred~~

~~While it wishes~~

**6.What character ends all strings?**

'.'

' '

'\0' \*

'\n'

**x7.Which of the following reads in a string named x with one hundred characters?**

fgets(x, 101, stdin);\*

~~fgets(x, 100, stdin);?~~

~~readline(x, 100, '\n');?~~

~~read(x);?~~

Top of Form

**8.Which of the following is the proper keyword to allocate memory in C?**

~~new~~

malloc()\*

~~create~~

~~value?~~

**9.Which of the following accesses a variable in structure \*b?**

~~~~b->var;\*

~~b.var;~~

~~b-var;~~

~~b>var;?~~

**10.Consider the following program fragment.   
d = 0;  
for(i = 1; i < 31; ++i)   
for(j = 1; j < 31; ++j)   
for(k = 1; k < 31; + +k)  
if ( ( (i + j + k) % 3) == 0)  
d = d+1;  
printf("%d",d);  
The output will be**

9000\*

~~27000?~~

~~3000~~

~~none of the above~~

Top of Form

**x11.Are the following declarations same?  
  
char far \*far \*scr;  
char far far\*\* scr;**

Yes

No \*

**12.Functions cannot return a floating point number**

Yes

No \*

:00:06:54

Top of Form

**x13.The statement in the previous question can never print**

~~one~~

two\*

~~three~~

~~four~~

**14.What will be the output of the program ?  
#include<stdio.h>  
#include<string.h>  
int main()  
{  
char str1[5], str2[5];  
int i;  
gets(str1);  
gets(str2);  
i = strcmp(str1, str2);  
printf("%d\n", i);  
return 0;  
}**

Unpredictable integer value\*

~~0~~

~~-1?~~

~~Error~~

**x15.If the two statements  
\* p v = 0;  
printf ("%d %d", \*pv, v);  
are appended to the previous program fragment then the output will be**

~~0 3?~~

~~~~0 0\*

~~unpredictable?~~

~~none of the above?~~

Bottom of Form

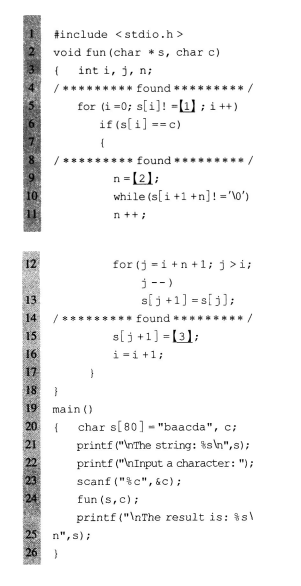
Bottom of Form

Bottom of Form

Bottom of Form

Bottom of Form

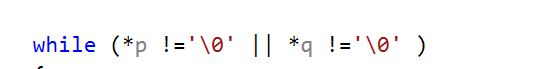
Bottom of Form

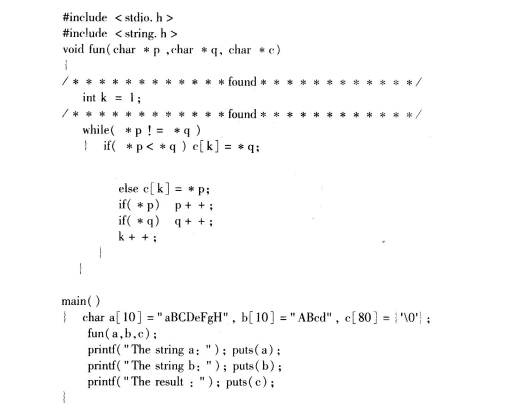


2. The function "fun" will try to compare the two strings in p and q from the first position to the last. The comparing criteria is to find the char which has larger ASCII value. It will put the comparision result in string c. If p and q don't have the same length, then only compare the part with same length, and put the rest of the longer string directly to c.

If p = "aBCDeFgH" q = "Abcd" . then c = "abcdeFgH" because 'a' > 'A', 'b' > 'B', 'c' > 'C', 'd' > 'D'

There are two line of code have bugs. please find them and correct them.

1   
  
C:\Users\Yu\Documents\Tencent Files\651412330\Image\C2C\`MILZW444]ZCOU{A37{KGA3.png  
  
  
  




char a[10] = "aBC", b[10] = "ABcdaaaaa", c[80] = {'a', '\0' };

fun(a, b, c);

printf("The string a: "); puts(a);

printf("The string b: "); puts(b);

printf("The string c: "); puts(c);

void fun(char \*p, char \*q, char \*c)

{

int k = 0;

while (\*p !='\0' || \*q !='\0' )

{

if (\*p < \*q)

{

c[k] = \*q;

}

else

{

c[k] = \*p;

}

k++;

if (\*p)

{

p++;

}

if (\*q)

{

q++;

}

}

}