速成教育
SPEED UP EDUCATION

CSCA48

Introduction to Computer Science II

导师： **VC**

2019W CSCA48 Midterm

Unit 1 – Fundamental concepts and general structure of C programs

1.- [2 marks] Read carefully the following piece of code, and ***complete the required variable declarations in the gray box at the top of main() so that the code would compile and run without unexpected behaviour. Make reasonable assumptions where needed.***

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
```

```
    x=5;
    y=x*x;
    a[5]=(char)x;
    a[6]='\0';
    printf("%f %d %s\n",y,x,a);
    return 0;
}
```

2.- [2.5 marks] Read the following code carefully and complete the function definition so that the ***code compiles and wun without enexpected behaviour. Missing parts are shown by '_____', complete all of these!***

```
#include<stdio.h>
#include<stdlib.h>

___________ BabylonianRoot(_________ S, __________ its)
{
    // Approximate the square root of a number using the Babylonain method

    _______________

    _______________
    xn=S;
    for (i=0; i<its; i++)
    {
        xn=(.5*(xn+(S/xn)));
    }
    return xn;
}

int main()
{
    float x=4.0;
    int trials=10;

    printf("The square root of %f is %f\n",x,BabylonianRoot(x,trials));
    return 0;
}
```

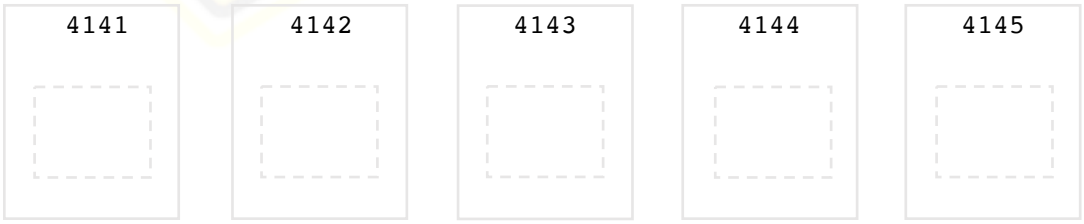
Unit 2 – Memory model, variables, pointers, and arrays

3.- [6 marks] Complete the code below, and ***show what happens in memory when the code is run up to the point where the function is about to return a value back to main().***

```
int quizAvg(float *q1, float *q2, float *q3)
{
    // Returns the average mark for the 3 quizzes

    // Complete the function in the space below (The answer is 1 line only!)

}

int main() {
    float quizzes[3]={95, 75, 85};
    float avgMark=0;

    // Complete the function call in the space below (1 line only!)


    printf("The average is %f\n", avgMark);
    return 0;
}
```

Variable for main() go about this line

Variable for quizAverage() go below this line



(Make sure you have filled in the ***contents*** of the boxes for each variable according to what the program does!)

Unit 3 – Organizing, Storing, and Accessing Data

4.- [3 marks] In the space below, provide the definition in C for a compound data type that will be used to store information about PeekeMons (fantastically powerful beings).

PeekeMons have the following attributes (field names correspond to attribute names):
- `name` (which should be cool-sounding)
- `age` (in years, PeekeMons can live for up to 1000 years)
- `hunger` (from 0 - not hungry, to 100 - will eat the next thing you place near it!)
- `picture` (a char array of size 256x256 with the black&white image of the PeekeMon)
- `voice` (a floating point array of size 88200 x 2, containing a 2-second digital recording of its voice)

```
// Provide here the definition for the compound data type requested above
// The new data type must be named 'PeekeMon'
```

5.- [3 marks] Having travelled through the cosmos to find the most awesome PeekeMons, you have decided to find a conservation area where PeekeMons roam freely and are not endangered by unethical humans trying to let them involved in fighting contests. To keep track of the PeekeMons, you built a little app that stores imfornation for each of them using a linked list.

In the space below, provide the definition of the data type needed to store ***one node of the linked list.*** The node must contain ***data for one PeekeMon*** (in a field called `'creature'`), ***plus an array*** (call this field `'coordinates'`) ***with 3 entries to hold the (x,y,z) coordinates*** (you have to decide what's the appropriate type for these) where the PeekeMon can be found within the conservation area.

```
// Provide here the definition for the compound data type for the list node
// Name this data type 'PeekeMonNode'
```

Unit 3 – Organizing, Storing, and Accessing Data

6.- [3 marks] You have been reviewing the recordings of the PeekeMon voices for the pleasures in your biolgical preserve, and you discover that for one of them, the recording contains corrupted data that needs to be cleaned up by hand.

As noted before, we have a 2-second recording, stored within an array of size 88200 x 2 called 'voice', That means ***there are 88200 digital samples***, and 2 channels (left and right). So if you want to access ***the sample*** (digitally recorded value) at position 451, you would get:

`left_channel` is stored inside the voice array at `voice[451][0];`
`right_channel` value is stored in the voice array at `voice[451][1];`

These are floating point values in [-1, +1], and if you send this array to the sound card in your computer, it will play the sound this creature makes!

You have determined ***that the samples to be fixed*** have to be fixed (set to zero) by hand are at ***indexes***:

```
23581
48724
57623
```

And you need to reset to zero both the left and right channel for these samples.

Complete in the space below the code you'd need in order to update these samples to have a value of zero for both the left and right channel; given the creature variable defined in main().

```
int main()
{
    PeekeMon achooPic;
    // Assume we have filled in the data for this creature
    // Write below the code to update the requested samples









}
```

Unit 3 – Organizing, Storing, and Accessing Data

7.- [3 marks] While solving the above, you realize it may be useful to have a litle function that can be used to change the name of a PeekeMon (because you plan to start an adopt a PokeeMon program to get funds for the reservation). Complete the code below so that the creature's name is changed in main() to something you like!

```
void PK_name_change(____________________, char new_name[1024])
{
    // Update the name of this PeekeMon in the space below. 1 line only!




}
int main()
{
    PeekeMon achooPic;

    // Assume we have filled-out the data for this PeekeMon
    // Complete the call to change the creature's name, 1 line only!




}
```

8.- ***Ronus 1 mark - DRAW the most awesome PeekeMon you have found in your travels, and give it a name!***

Unit 3 – Organizing, Storing, and Accessing Data

9.- [6 marks] Every now and then, PeekeMons get adopted and taken off with servation (of course, you first do a thorough check that the people adopting are responsible and will take good care of the creature!). Suppose you have a ***search function that returns a pointer to the node in the list for a creature matching the name provided to the search function.***

Complete in the space below the code for the ***delete_PK()*** function that removes this creature's node from the linked list.

```
PeekeMonNode *delete_PK(PeekeMonNode *head, PeekeMonNode *del)
{
    // Remove from the linked list the
    // the node pointed to by 'del' creature whose data is contained in

    // Add here any variable declarations you will need for this function
    // If you will declare any pointers, please call them p, q … etc.

    // Add here any code that should be run before the loop below

    while( _____ )  // Complete the while loop condition!
    {
    // Add code here to complete the removal process (no more than 7-8 lines with
    // curly braces included)

    }
    // Something needs to be returned, maybe?

}
```

Unit 3 – Organizing, Storing, and Accessing Data

10.- ***CRUNCHY*** - It's feeding time for the PeekeMons! You need to know which of them are super hungry so you can feed those first (this will help avoid the situation where a very very hungry creature eating up trees and plants in the conservation area).

You decide to write a function that will ***return a list of the PeekeMons whose hunger level is greater than a given threshold.***

However: You **do not want to duplicate PeekeMon data at all.**
(this means you should not make copies of PeekeMon variables or PeekeMonNodes)

10.1 - [5 marks] ***In no more than 5 steps, describe the process you will implement to solve this task***

10.2 - [5 marks] ***DRAW below a diagram that illustrates how the list returned by your function works*** - The diagram should show how the list returned by your new function provides access to the data for the PeekeMons which is you know is also in a linked list (be sure to indicate which list is which).
You're explaining how your function works, your diagram has to be understandable receive any marks.
You can practice and try things out at the back of the page.

Mutiple Choice Questions

MPORTANT NOTES:
- You can scratch and mark in the questions below ***but please mark just one answer in the bubble-sheet at the back of the test. You will not get marks for multiple scratches.***
- ***Do not forget to record your answers*** - It's strongly recommended you mark your answers as you solve this section to avoid problemns.
- Think carefully - each question is intended to test whether you have fully understood a particular idea or have learned a specific concept from the past 6 weeks.

1.- Which of the following ***is not true*** according to the memory model we have been using for programming in C?

a) Each variable has its own box b) Parameters get their own box

c) A box can have more than one name tag d) Boxes in memory have different sizes

e)The locker number where data is stored doesn't matter

2.- We use explicit type-casting because

a) The compiler doesn't do typecasting automatically b) Variables have fixed data types

c) Without it our code won't compile d) It helps avoid bugs

e) It saves computations

3.- One of the following options ***must be true*** after we execute the code `'y = (char) x'`

a) 'x' will be of type 'char' b) 'y' will have a value equivalent to the 'char' value of 'x'

c) 'y' will be of type 'char' d) 'y' and 'x' must have the same value

e) None of the above is certain to be true

4.- The reason why we have different numeric types for integers and for floating point numbers is:

a) The CPU has separate circuits for these numeric types

b) The float' variables require more memory because they have more digits

c) It's a quirk of the C language, since we know in Python we can use them interchangeably

d) It's a convention that programmers use to make code more structured and easier to debug

e) They have to be processed differently by printf()

5.- Which of the following options is ***not a potential result*** of changing data ***outside the bounds of an array.***

a) Your program crashes b) The program runs but produces weird/incorrect output

c)The value of other data in your program may change d) Nothing happens

e) The compiler throws out an error when your code is running

---

6.- In tutorial, your TAs showed you how to allocate an array of a particular size ***on-demand*** using ***calloc().*** Suppose you use this to allocate space for a new string (a char array) with capacity for 1024 characters, and get a pointer '***p***' to your new array. You then store into that string the text "HELP". What is the value of `'*(p+4)'`?

a) 'P' b) The locker numeber for the string, plus 4 c) \0'

d) We can't tell since the string can be anywhere in memory

e) 'junk' since memory is full of it before we assign it a value

---

7.- Your message from (6) is missing emphasis, so you decide to add these two lines to your program:
(Think carefully about memory contents given how your string was allocated in 6 before answering).

```
for (int i=4; i<10; i++)
{
    *(pti)='!';
}
printf("%s\n",p);
```

What happens when you compile and run your code?

a) It works correctly and prints "HELP!!!!!!" b) The program crashes!

c) The program prints "HELP!!!!!!" followed by junk d) It doesn't compile, there's an syntax error

e) It compiles but the compiler detects and prints an error when the code is running

---

8.- Which of the options below is something we can not do with compound data type variables?

a) Create an array with many of them b) Pass them as parameters to functions

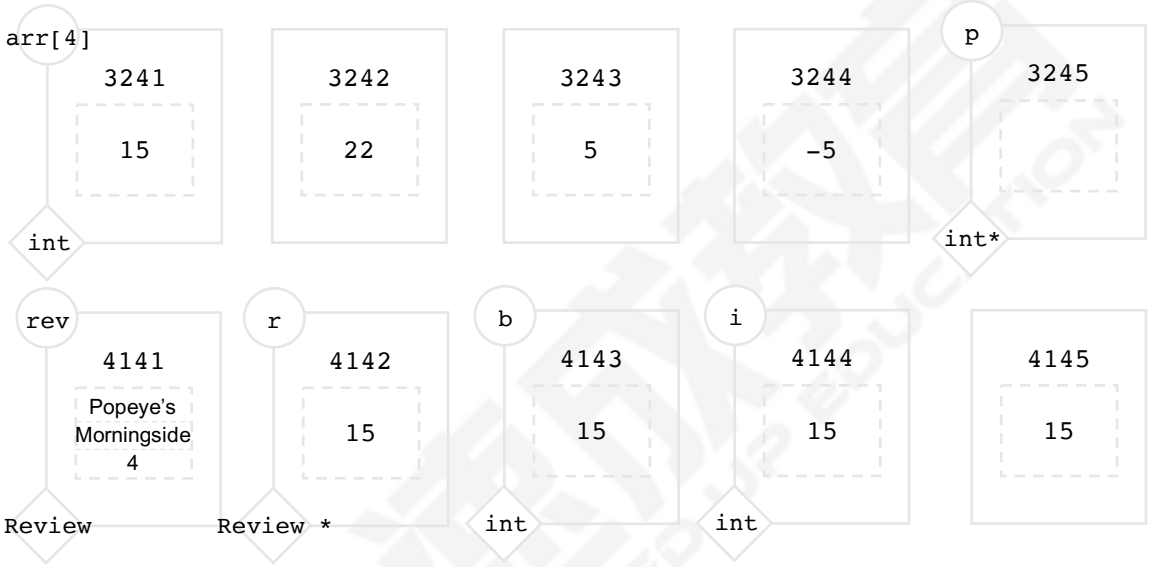
c) Use them in a for loop d) Get a pointer to individual fields within the variables

e) Create them on-demand

10.- Suppose that yuo're given the task of implementing a little database for an on-line book/music store called ***Aquamarine***. The store provides interactive touch-screens where customers can search for books or music that they want, using keywords. There is a ***master link list*** that contains all the information regarding every book and music album the store sells, and it is a requirement that the list be kept in sorted order by the title of the book or album. Which of the functions below needs to be carefully implemented in order to implement this functionality?

a) Search b) Delete c) List traversal

d)strcmp() e) Insert



11.- In the diagram above, ***what is the value of 'p' after*** `p=&arr[0] + 2;`

a) 37 b) 3243 c) 17 d) 22 e) 3247

12.- Suppose that `p=&arr[0];` What are the correct instructions to swap the 2nd and 4th entries in the array?

a)
```
*(p+1)=*(p+3);
*(p+3)=*(p+1);
```

b)
```
i=p+1;
p+1=p+3;
p+3=i;
```

c)
```
i=*(p+3);
*(p+3)=*(p+1);
*(p+1)=i;
```

c)
```
i=*(p)+3;
*(p)+3=*(p)+1;
*(p)+1=*i;
```

e)
```
*(p++)=*(p+3);
*(p+3)=*(p--);
```

13.- Suppose that r=&rev; and p=&arr[0]; #455 12 of 13
What instruction updates the score field in the review variable to '5'?

```
a) *(r)->score=*(p)+2;        b) r->score=*(p+2);            c) r.score=5;

    d)  rev->score=*(p+2);                    e) rev->score=5;
```

14.- Suppose that p=&i; what is the ***value*** of ***arr[1]*** after we run the following code:

```
for (*(p)=0; *(p)<4; (p)++)
{
  arr[*(p)]=*(p);
}
```

a) Junk b) The code above won't compile c) 4145 d) 4144

15.- Which of the following instructions is a ***bug***? Think ***very carefully*** abour what the instruction actually does in the memory model! Assume `p=&i;` and `r=&rev;`

```
a) b=p;                    b) p=(int *)&rev;              c) *(r+1)=*(r);

                  d) r->score=*(p);            e) *(p)=i;
```

That's it for your midterm! :) please make sure you recorded all your answers in the bubble-sheet page