

SCHOOL OF SCIENCE & MEDICINE

Exam Title: **PROBLEM SOLVING AND PROGRAMMING 2**

SPRING 2018

EXAMINATION FOR THE DEGREE PROGRAMMES IN

BSC Computing September 2017 and January 2018

Exam Code: 20182AIAP113AWP

TIME ALLOWED: **2 HOURS**

MATERIALS PERMITTED: None

MATERIALS PROVIDED: Word sheet will be provided electronically
in the exam folder

INSTRUCTIONS:

1. Answer ALL FOUR questions, for question 1 and 3 you need to answer only one section from each
2. This examination requires both written work in the answer book as well as practical work on the computer.
3. Be tidy with your written work. Marks may be deducted for untidy work.
4. You must save your practical work on the designated disk drive on the desktop of your computer.
5. You must organise your answers into folders, one folder for each question. Do not provide more than one answer to each question.
6. You may use the file "ExamReferencePSP2" on the LiveExamMaterial folder for a quick reference of some aspects you might need to solve the questions.

Question 1 (practical) (25 marks)**Answer either section 1 or section2****Section 1:**

A. Create a Flower structure to present type of flowers as below:

Name: (e.g. Rose, Tulip, BlueBell, Dalia, etc.)
Season: (e.g. Spring, Summer, ...)
Colour: (e.g. White, red, ...)
Height: (number in centimetres)
Age: (number of weeks)

B. Create two flowers variables and populate a proper data in them.

C. Write a function to print the content of one structure to the console.

D. Create an array of FIVE Flowers and populate four flowers in the array.

E. Write a function that print out the flowers with a height more than a certain value.

(A:5, B:5, C:5, D:5, E:5) marks

Section 2:

Using the two arrays shown below, write a C program project:

ArrayA={0, 1, 2, 3, 4, 5} ArrayB={3, 4, 5, 6, 7,8}

A. Populate the data shown above in the two arrays in your program.

B. Write a function called FIND, this function used to verify if a number is belong to the array or not.

C. Write a function called INTERSECT that can populate the intersected values from the two arrays above and print them to the console.

i.e. The intersection output should be: {3, 4, 5}

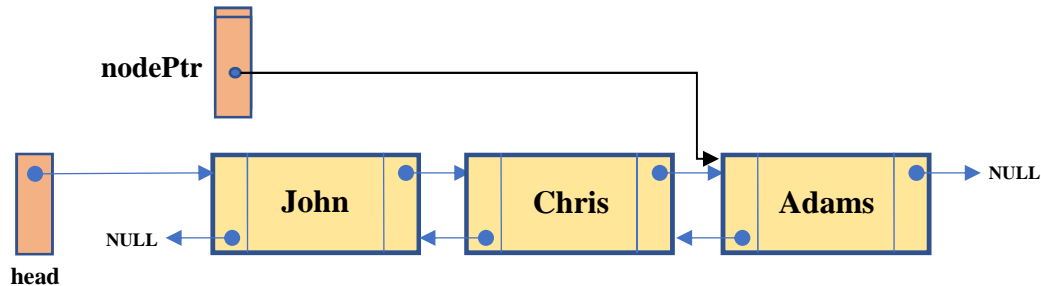
D. Write a function called UNION that can print to the console the values of both arrays with no duplicate.

i.e. The Union output should be: {0, 1, 2, 3, 4, 5, 6, 7, 8}

(A:6, B:6, C:6, D:7) marks

Question 2: (Written) (25 marks)**Answer either section 1 or section2****Section 1:**

Using figure 1 below, check the provided code below by marking the wrong statements (as applicable) and show your correction(s).

**Figure 1****A. To insert a New node with “Alex” between Chris’s node and Adams’s node.**

```

New-> name="Alex" ;
New-> next=nodePtr->next ;
New-> prev=nodePtr->prev ;
head-> next->next-> New;
nodePtr->prev=New;
  
```

B. To insert a New node with “Lina” at the beginning of the list.

```

New-> name="Lina" ;
New-> next=head ;
head = null;
New-> prev=head ;
head = New;
  
```

C. To delete Chris’s node:

```

nodePtr->prev = head ;
Head->next = nodePtr ;
nodePtr->prev = null;
  
```

D. To delete Adams’s node:

```

Head->next->next = nodePtr->next ;
nodePtr = null ;
  
```

(6 to 7 marks each)

Section 2:

Follow the program code below, fill the memory locations provided as appropriate, re-write the code and the memory tables filled with data in your answer book.

```
// Code for question 2 section 2
```

```
int x, *px1, *px2 ;
```

```
double d, *pd1, *pd2 ;
```

```
int main ()
```

```
{
```

```
    x= 5 ;
```

```
    px1 = &x ;
```

```
    px2 = new int ; *px2 = 12 ;
```

```
    d = *px2 + 20 ;
```

```
    pd1 = &d ;
```

```
    pd2 = new double ;
```

```
    *pd2 = *pd1 + 1.1 ;
```

```
    return 0 ;
```

```
}
```

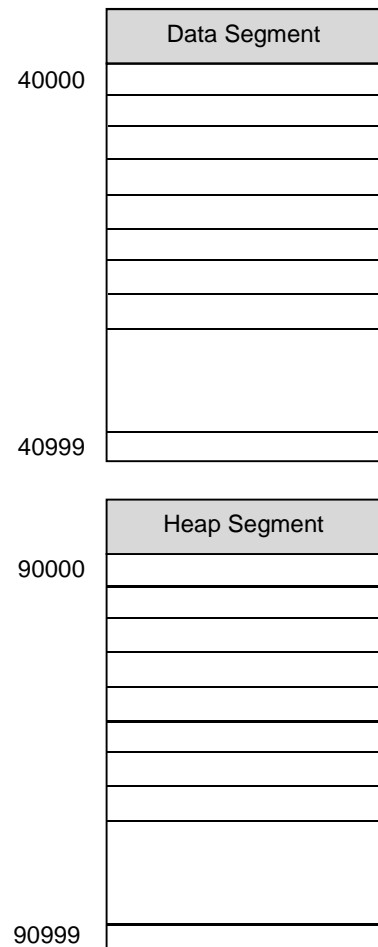


Figure 2

- Fill the memory locations in the segments as appropriate with data (numbers or addresses).
- Make sure that you are using the right segment for each variable and data location.
- Make sure that you are allocating the right size for each data or address, remember that each location in the memory is of one-byte length, which is represented by a single box (row) in the figure 2 above.

NB: the address used in this question are decimal for simplicity, do NOT used hexadecimal.

(A:15, B:5, C:5) marks

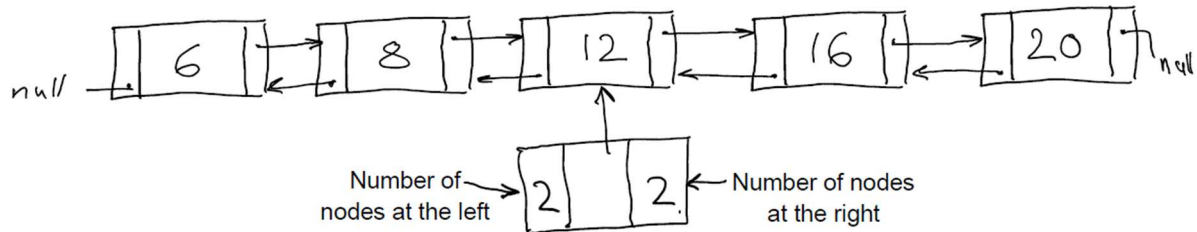
Question 3 (written) (25 marks)**Answer either section 1 or section2****Section 1:****Figure 3**

Figure 3 above represents a sorted doubly linked list, this linked list is maintained using a pointer pointing to the middle node of the list, this pointer is included in a structure called MidNode, this structure has a pointer to the middle node of the doubly linked list, and two integers. The left integer indicates on how many nodes are at the left of the middle node and the right integer indicates on how many nodes are at the right of the middle node.

In your answer book answer the following:

- Write a code to define the doubly linked list structure.
- Write a code to define the MidNode structure.
- Write the insert function, this function will maintain the insertion to the left or the right based on the inserted node value, for example if the user would like to insert a node with value 18, the insertion should go the right between node 16 and 20.
- Make sure in the function in 'C' to maintain the pointer to be pointing to the middle node when multiple insertion is performed.

(A:6, B:6, C:8, D:5) marks

Section 2:

Using the binary tree in Figure 4 below, answer the following:

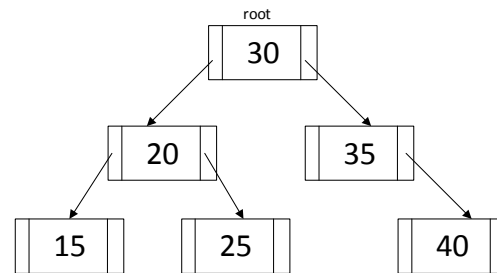


Figure 4

Use the original tree in figure 4 each time you answer each of the questions below:

- After adding two nodes containing 32 then 34 to the binary tree above, is the tree become a balanced tree? Explain why?
- Explain the needed steps to delete the node containing 35.
- Explain the necessary steps to update the node with value 30 to 14.
- Write a C function that can traverse all the nodes in the tree above and output all the values in a sorted way starting from the smallest to the highest.
- Write a C function that can be used to search for a value in the tree.

(A:5, B:5, C:5, D:5, E:5) marks

Question 4 (written) (25 marks)

Calendars became one of the essential application used on daily bases. As you know a typical calendar has the following inputs:

1. Event name
2. Date
3. Time
4. Duration
5. Location
6. People Invited (could none and up to 3 max)

In this question you need to suggest a data structure that hosts the contain of the Calendar in an efficient way and makes the displaying and the retrieving of the calendar very easy.

To answer the question, you need to do the following:

- A. Write the needed code to present the calendar with the above inputs.
- B. Write the steps needed to input a certain event to your calendar.
- C. Discuss your answer based on the efficiency and the space used to save and retrieve data in/from memory.

(A:8, B:8, C:9) marks

- END OF PAPER –