**Module Assignment 5**

Complete Module Assignment 5 when you finish Learning Package 12.

There are 20 questions in this assignment. All questions carry equal marks. There is only one correct answer, and you should select the option you consider the most appropriate.

Write down the missing word or phrase:

1. The keyword **t f** enables new names for data types to be created. This can be useful for improving the readability of the program.
2. Structures enable the programmer to group variables together, even those of **d t** data types, under a single variable name.
3. It is possible to have **a s** of the newly defined structures.
4. Unless the variables are declared at the same time as the structure template, both the keyword ***s t*** and the tag name are needed when declaring new variables.
5. The *typedef* keyword can be put before the keyword *struct* as part of the **d n** to shorten the process.
6. A structure can also contain other **s es**.
7. The -> operator is the **m r a s** operator and is used to ‘get to’ a member of the structure via a pointer.
8. A **l d l t** is a data structure where can be expanding by adding another element.

9. A linked list element has two parts one with the data to be stored, and a second part that holds where the next **e t** to be connected is.

1. We need some way of **a e** memory each time we want a new element.
2. A function that calls itself is a **r e** function.
3. A stack can be thought of as a pile of items, one stacked on top of the other. If you want to put an item on top the stack, you **p h** it on at the top of the stack.
4. If you want to take something off the stack it has to be taken off the

**t\_p** of the stack.

1. We can put new elements on to the stack by creating a new element, setting the next pointer to in the element that is currently at the top of the stack and change the pointer in the stack structure to **p t** to the new element.
2. The new element is now the **t\_p** of the stack.

Question 16-20 relate to windows programming:

1. The function **M eB\_x** has four parameters.
2. The first parameter is a handle to a parent window, which is often a

**n l**.

1. The second parameter is a pointer to a string that will form the

**m e** inside the window.

1. The third parameter is a **p r** to a string that puts a message in the windows title bar.
2. The last parameter is used to specify the **s e** of the window.

**Answers**

1. The keyword **typedef** enables new names for data types to be created. This can be useful for improving the readability of the program.
2. Structures enable the programmer to group variables together, even those of **different** data types, under a single variable name.
3. It is possible to have **a s** of the newly defined structures.
4. Unless the variables are declared at the same time as the structure template, both the keyword ***struct*** and the tag name are needed when declaring new variables.
5. The *typedef* keyword can be put before the keyword *struct* as part of the **defintion** to shorten the process.
6. A structure can also contain other **structures**.
7. The -> operator is the **member access** operator and is used to ‘get to’ a member of the structure via a pointer.
8. A **linked list** is a data structure where can be expanding by adding another element.
9. A linked list element has two parts one with the data to be stored, and a second part that holds where the next **element** to be connected is.
10. We need some way of **address/access** memory each time we want a new element.
11. A function that calls itself is a **recursive** function.
12. A stack can be thought of as a pile of items, one stacked on top of the other. If you want to put an item on top the stack, you **push** it on at the top of the stack.
13. If you want to take something off the stack it has to be taken off the

**top** of the stack.

1. We can put new elements on to the stack by creating a new element, setting the next pointer to in the element that is currently at the top of the stack and change the pointer in the stack structure to **point** to the new element.
2. The new element is now the **top** of the stack
3. The function **MessageBox** has four parameters.
4. The first parameter is a handle to a parent window, which is often a

**null**.

1. The second parameter is a pointer to a string that will form the

**message** inside the window.

1. The third parameter is a **pointer** to a string that puts a message in the windows title bar.
2. The last parameter is used to specify the **style** of the window.