# Homework 5 Subroutines Answers

# 1. Describe briefly three advantages of using subroutines in programs. [6]

Large programs are broken down into subtasks that are easier to program and manage

Each module can be individually tested and debugged

Modules can be re-used several times in a program

Frequently used modules can be saved in a library and used by other programs

Several programmers can simultaneously work on different modules, shortening development time

A well-organised, modular program is easier to follow

If the program needs to be modified, it is easier to find which module needs to be changed

Self-contained modules mean that the change should not affect the rest of the program

New features can be added by adding new modules

# 2. (a) What is a global variable? [1]

 A variable that has scope anywhere in the program, including subroutines, without having to be passed as a parameter.

#  (b) What is a local variable? [1]

 A local variable only has scope within the routine in which it is used; it can’t be accessed elsewhere without being passed. The memory location is no longer available after the routine has finished running.

#  (c) What is the advantage of using local variables in subroutines? [2]

#  Using local variables keeps the subroutine independent of the main program, so that it can be used without inadvertently affecting any variables in the main program.

# 3. Write a program in pseudocode that allows the user to add a name to a list array that holds up to ten names. The main program should call a subroutine that presents the user with a menu of three choices:

#  1 Add name

#  2 Display list

#  3 Quit

#  The subroutine accepts and validates the user choice (1-3). If it is invalid, an appropriate error message should be displayed and the user asked to re-enter until they input a valid choice.

 The subroutine is called with the statement

 choice = displayMenu()

The main program then branches to one of two subroutines which adds a name or displays the list, or quits the program with a message “Program terminating” if the user selects option 3.

#

#  If the user chooses to add a name, they should be prompted to enter a list number indicating where they want the name to be inserted. If the location is occupied, it will overwrite the name.

#  The program will also provides an option to display the list. The output should look something like the example below. [10]



 Total 20 marks

#

SUB displayMenu()

 OUTPUT (“1 Add Name”)

 OUTPUT (“2 Display list”)

 OUTPUT (“3 Quit”)

 choice 🡨 USERINPUT

 WHILE choice < 1 OR choice > 3

 OUTPUT (“Error… choice must be 1 to 3: ”)

 choice 🡨 USERINPUT

 END WHILE

 RETURN choice

ENDSUB

SUB addName(names)

 OUTPUT (“Enter name: ”)

 newName 🡨 USERINPUT

 OUTPUT (“Enter position in the list to insert name (1-10): ”)

 index 🡨 USERINPUT

 names[index] 🡨 newName

ENDSUB

SUB displayNames(names)

 FOR index 🡨 1 TO 10

 IF names[index] NOT empty

 OUTPUT (names[index])

 END IF

 END FOR

ENDSUB

#main program

declare array names[1:10]

choice 🡨 0

WHILE choice <> 3

 choice 🡨 displayMenu()

 IF choice = 1 THEN

 addName(names)

 ELSE IF choice = 2 then

 displayNames(names)

 END IF

END WHILE

OUTPUT “Program terminating”