PAST PRE-RELEASE

The pre-release material will arrive a few months before your examination because it is different for each examination series.
Here is a checklist of useful things to do.

1. Read through the pre-release material several times. Check with your teacher if there is anything at all that you do not understand.
2. For each task, write an algorithm using both pseudocode and a flowchart to show what is required.
3. Choose sets of test data that you will need to use, and work out the expected results. Remember to use normal, boundary and erroneous data. Be able to give reasons for your choice of test data.
4. Complete trace tables to test your pseudocode and flowcharts. This will enable you to ensure that both the pseudocode and the flowcharts work properly. It is a good idea to get another student to trace your algorithms as well.
5. Decide which works best for each task, pseudocode or a flowchart, and why.
6. Before starting to write your program for each task:
    a. Decide the variables, including any arrays, and constants you will need
    b. Decide the data types required for these
    c. Decide the meaningful names you will use
    d. Be able to explain your decisions.
7. If you are asked to repeat the same thing many times, for example adding up totals, complete the task for one and check it works before repeating it many times.
8. Write and test each task. You can use the same test data as you used for your pseudocode and flowcharts.

MUHAMMAD yousuf mateen

MAY/JUNE 2015

Q 1 You are advised to spend no longer than 40 minutes answering this section. Here is a copy of the pre-release material.

DO NOT attempt Tasks 1, 2 and 3 now.

Use the pre-release material and your experience from attempting the tasks before the examination to answer Question 1.

Pre-release Material

Write and test a program to complete the **three** tasks.

TASK 1

A data logger records the temperature on the roof of a school twice a day, at midday and midnight. Input and store the temperatures recorded for a month. You must store the temperatures in two one-dimensional arrays, one for the midday temperatures and one for the midnight temperatures. All the temperatures must be validated on entry and any invalid temperatures rejected. You must decide your own validation rules. You may assume that there are 30 days in a month.

TASK 2

Calculate the average temperature for midday and the average temperature for midnight. Output these averages with a suitable message for each one.

TASK 3

Select the day with the highest midday temperature and the day with the lowest midnight temperature. Then output each of these temperatures, the corresponding day and a suitable message.

Your program must include appropriate prompts for the entry of data. Error messages and other outputs need to be set out clearly and understandably. All variables, constants and other identifiers must have meaningful names. Each task must be fully tested.

1.

a. All variables, constants and other identifiers should have meaningful names.

i. In **Task 1**, you had to store the midday temperatures and midnight temperatures in arrays.
Write suitable declarations for these **two** arrays.

.....

.....[2]

ii. It has been decided to record the temperatures for one week rather than one month.

Write the new array declarations that you would use.

.....

.....[1]

iii. Declare **two** other variables that you have used and state what you used each one for.

Variable1…….....

Use…….....

Variable 2.....

Use……..……..[4]

MUHAMMAD yousuf mateen

b. Write an algorithm to complete **Task 2**, using **either** pseudocode, programming statements or a flowchart. Use temperatures for **one week** only. You should assume that Task 1 has already been completed.

MUHAMMAD yousuf mateen

……[5]

c. Give a set of midday temperature data, for a week, that could be used to check your validation rules for Task 1. Explain why you chose this data set.

Data set……

……

Reason for choice

……

……

……[2]

d. Explain how you select the day with the highest midday temperature (part of **Task 3**). You may include pseudocode or programming statements as part of your explanation.

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……[6]

MUHAMMAD yousuf mateen

MAY/JUNE 2015

Q 2 You are advised to spend no longer than 40 minutes answering this section.

Here is a copy of the pre-release material.

DO NOT attempt Tasks 1, 2 and 3 now.

Use the pre-release material and your experience from attempting the tasks before the examination to answer Question 1.

Pre-release Material

Write and test a program to complete the **three** tasks.

TASK 1
A school keeps records of the weights of each pupil. The weight, in kilograms, of each pupil is recorded on the first day of term. Input and store the weights and names recorded for a class of 30 pupils. You must store the weights in a one-dimensional array and the names in another one dimensional array. All the weights must be validated on entry and any invalid weights rejected. You must decide your own validation rules. You may assume that the pupils' names are unique. Output the names and weights of the pupils in the class.

TASK 2
The weight, in kilograms, of each pupil is recorded again on the last day of term. Calculate and store the difference in weight for each pupil.

TASK 3
For those pupils who have a difference in weight of more than 2.5 kilograms, output, with a suitable message, the pupil's name, the difference in weight and whether this is a rise or a fall.
Your program must include appropriate prompts for the entry of data. Error messages and other outputs need to be set out clearly and understandably. All variables, constants and other identifiers must have meaningful names. Each task must be fully tested.

1.

a. All variables, constants and other identifiers should have meaningful names.

i. Declare the array to store the pupil's names

……[1]

ii. Declare the array to store the pupil's weight.

…… [1]

iii. It has been decided to record the weight for the whole school of 600 pupils rather than one class.
Write suitable new declarations for these two arrays.

……

…… [1]

b. Write an algorithm to complete **Task 2,** using **either** pseudocode, programming statements or flowchart. Use weights for the whole school. You should assume task 1 has already been completed.

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

MUHAMMAD yousuf mateen

……

……[5]

c.

i. Describe suitable variable rules for Task 1.

……

……

……

……[2]

ii. Give **two** pupil weights that you could use to check the validation used in **Task 1.**

Explain why you chose each weight.

Weight 1 ……

Reason for choice ……

……

Weight 2 ……

Reason for choice ……

…… [4]

iii. Explain how you select the pupils with a fall in weight of more than 2.5 kilograms (**part of Task 3**). You may include pseudocode or programming statements as part of your explanation.

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……[4]

MAY/JUNE 2016

Q 3 **Pre-release Material**

The manager of a building materials delivery service needs a program to check the contents and weight of sacks to ensure that correct orders are made up for delivery. A price for the order will be calculated.
Write and test a program for the manager.

Your program must include appropriate prompts for the data entry

- ✓ Error message and other output need to be set out clearly
- ✓ All variables, constants and other identifiers must have meaningful names
- ✓ You will need to complete these three tasks. Each task must be fully tested.

Task 1- Check the contents and weight of a single sack.
Each sack must obey the following rules to be accepted.

- ✓ Contain cement, gravel or sand, with a letter on the side for easy identification.
  - C-Cement
  - G-Gravel
  - S-Sand
- ✓ Sand or gravel must weight over 49.9 and under 50.1 kilograms
- ✓ Cement must weight over 24.9 and under 25.1 kilograms

Input and store the number and contents for one sack. The contents must be checked and incorrect sack rejected. The weight must be validated on entry and an overweight or underweight sack rejected.
Output the contents and weight of an accepted sack. It a sack is rejected, output the reason(s).

Task 2-Check a customer's order for the delivery

Input and output the number of sacks of each type required for the order. Use Task 1 to check the contents and weight of each sack. Ensure that the delivery contains the correct number and type of sacks for the order.
Output the total weight of the order.
Output the number of sacks rejected from the order.

Task 3- Calculate the price of the customer's order. Prices for the sacks are as follows.

- ✓ Regular price for each sack
  - Cement, $3
  - Gravel, $2
  - Sand,$2
- ✓ Discount price for a special pack containing 1 sack of cement, 2 sacks of sand and 2 sacks of gravel, $10

Calculate and output the regular price for the order. Check how many special packs are in order. If a discount price applies then output the new price for the order and the amount saved.

1.

a. All variables, contents and other identifiers should have meaningful names.

i. For **three** of the variables that you have used in **Task 2,** state the name, type and its use.

Variable 1 name ……..

Type ……..

Use ……

Variable 2 name ……..

Type ……..

Use ……

Variable 3 name ……..

Type ……

Use ……[3]

ii. State three constants that you could have used for **Task 1**. Give the value that would be assigned to each one.

Constant 1 name ……

Value 1……..

Constant 2 name ……

Value 2 ……..

Constant 3 name ……

Value 3 ..…….. [3]

MUHAMMAD yousuf mateen

b. Write an algorithm to complete **Task 2**, using either pseudocode, programming statements or a flowchart. You can assume that the weight and contents have already been checked and accepted in **Task 1**. You do not need to output the number of sacks rejected.

MUHAMMAD yousuf mateen

……
……[5]

c.

i. Give two different data values that could be used to check your validation rules for sand in **Task 1**. Explain why you chose each value.

Sand data value 1 ……
Reason for choice ……
……
Sand data value 2 ……
Reason for choice ……
…… [2]

ii. Give two different data values that could be used to check your validation rules for cement in **Task 1**. Explain why you chose each value.

Cement data value 1 ……
Reason for choice ……
……
Cement data value 2 ……
Reason for choice ……
…… [2]

d. Explain how your program calculates the price for an order (**Task 3**). You may include programming statements as part of your explanation.

……
……
……
……
……
……
……
……
……
……
……
……

.....

.....

.....

..... [5]

MAY/JUNE 2016

Q 4 You are advised to spend no longer than 40 minutes answering this section.

Here is a copy of the pre-release material.

DO NOT attempt Tasks 1, 2 and 3 now.

Use the pre-release material and your experience from attempting the tasks before the examination to answer Question 1.

Pre-release Material

The manager of a parcel delivery service needs a proram to check the size and weight of parcels to determine which parcels can be accepted for delivery.
Write and test a program for the manager.
Your program must include appropriate prompts for the entry of data.
Error messages and other output to be set out clearly.
All variables, constants and other identifiers must have meaningful names.
You will need to complete these **three** tasks. Each task must be fully tested.

TASK 1- Check the size and weight of a single parcel

Each parcel must obey the following rules to be accepted for delivery:

- Each dimension must be no more than 80cm
- The sum of the three dimensions must be no more than 200cm
- The weight of the parcel must be between one and ten kilograms inclusive.

Input and store the weight and dimensions for one parcel. All the dimensions and the weight must be validated on entry and an unsuitable parcel rejected.
Output if the parcel is accepted or rejected. If rejected, output **all** the reasons why the parcel was rejected.

TASK 2- Check a customer's consignment of parcels

Input and store the number of parcels in the consignment. Calculate the number of parcels accepted and the total weight of the parcels accepted. For each parcel that was rejected, output **all** the reasons why that parcel was rejected.
Output the number of parcels accepted and the total weight of parcels accepted.
Output the number of parcels rejected.

TASK 3- Calculate the price for a customer's consignment of parcels
Extend TASK 2 also calculate the price for each parcel, using the following rules:

1 kg to 5 kg inclusive costs $10 Each 100 grams over 5 kg, up to 10 kg, costs an extra $0.10 Your output should also include the price for each parcel accepted and the total price of the consignment.

1.

a. All variables, contents and other identifiers should have meaningful names.

i. For each of the variables that you have used to record dimensions of the parcels in **Task1**, state the name, types and its use.

Variable 1 name ……..

Type ……..

Use ……..

Variable 2 name ……

Type ……..

Use ……..

Variable 3 name ……

Type ……

Use ……[3]

ii. State two constants that you could have used for **Task 1**. Give the value that would be assigned to each one.

Constant 1 name ……..

Value 1……

Constant 2 name ……..

Value 3 …….. [3]

b. Write an algorithm to complete **Task 1**, using **either** pseudocode, programming statements **or** a flowchart.

……

……

……

……

……

……

……

……

……
……
……
……
……
……
……
……
……
……
……
……
……[5]

c. Give three different data sets that could be used to check your validation rules for task 1. Explain why you chose each data set.

Data set 1 ……
Reason for choice ……
……
Data set 2 ……
Reason for choice ……
……
Data set 3 ……
Reason for choice ……
…… [6]

d. Explain how your program calculates the price for a consignment (part of **Task 3**). You may include programming statements as part of your explanation.

……
……
……
……
……
……
……

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[4]

MUHAMMAD yousuf mateen

MAY/JUNE 2017

Q 5 **You are advised to spend no longer than 40 minutes answering this section.**

Here is a copy of the pre-release material.

DO NOT attempt Tasks 1, 2 and 3 now.

Use the pre-release material and your experience from attempting the tasks before the examination to answer Question 1.

Pre-release material

A teacher is planning a school trip to a theme park at the end of term. You have been asked to write a program to work out the cost per student, to record those who are going and whether they have paid. The maximum number of students who can go on the trip is 45.

Write and test a program for the teacher.

- Your program must include appropriate prompts for the entry of data.
- Error messages and other output need to be set out clearly.

All variables, constants and other identifiers must have meaningful names.

You will need to complete these **three** tasks. Each task must be fully tested.

TASK 1 – Work out the cost.

The cost of the trip for each student is a share of the cost of a coach plus the cost of entry to the theme park. The total cost of the coach will be $550. The entry cost to the park is $30 for each student. The theme park gives one free ticket for every ten that are bought, which must be taken into consideration. Set up a program that:

- Stores the cost of the coach
- Stores the cost of an entry ticket
- Inputs the estimated number of students taking part, this must be validated on entry and an unsuitable entry rejected
- Calculates and outputs the recommended cost per student to ensure the trip does not make a loss.

TASK 2 – Record the students who are going and whether they have paid.

Input and store the names of the students who have asked to go on the trip up to the maximum number allowed. Input and store whether each student has paid. Enable printouts to be produced to show students who have not paid and those who have paid.

TASK 3 – Work out final costs.

Not all students will end up going on the trip, for example they might not have paid. Modify the program so that it gives overall totals for the costs charged and the amount of money collected. Output whether the school trip has made a profit or loss, or has broken even, and the amount of the final balance.

1.

a. All variables, constants and other identifiers should have meaningful names.

i. Name **two** constants you used for **Task 1** and state the value of each one.

Constant

1.....

Value.....

Constant 1.....

Value……..[2]

ii. Name **two** arrays you used for **Task 2** and state the purpose of each one.

Array 1.....

Purpose.....

.....

.

Array 2.....

Purpose.....

.....[2

]

b. Write an algorithm to complete **Task 1**, using **either** pseudocode, programming statements **or** a flowchart.

MUHAMMAD yousuf mateen

MUHAMMAD yousuf mateen

……[5]

c.

i. Name and describe one suitable validation check you used for Task 1, and one suitable validation check for Task 2. Each check MUST be different.

Task 1

Name…

Description…

…

Task 2

Name…

Description…

… [4]

ii. For each validation check in **part (c)(i)**, give an example of suitable test data and explain why you chose the test data.

Test data for **(c)(i) Task 1** …

Reason for choice…

……

……

Test data for **(c)(i) Task 2** …

Reason for choice …

……

……

d. Explain how your program calculates whether or not you have made a profit (**Task 3**). Any programming statements used in your answer must be fully explained.

……

……

……

……

……

……

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....[3]