SECTION A 20 MARKS (10 MULTIPLE CHOICE QUESTIONS, 2 marks each)

Please answer this section on the **mark-reading sheet** that you received (**not** in your answer book).
Choose one option for every question.

QUESTION 1

Suppose the following declarations appear in a C++ program:

```
float amount, forWaiter;
char service;
```

If the following function header is given:

```
float tip(float amountP, char serviceP)
```

which of the options below is a correct calling statement of the function `tip`?

1. `forWaiter = tip(float amount, char service);`
2. `tip(amountP, serviceP);`
3. `forWaiter = tip(123.66, 'A');`
4. `forWaiter = tip(amountP, serviceP);`
5. None of the above options is a correct calling statement.

QUESTION 2

Suppose the following declarations appear in a C++ program:

```
float afford, amount;
int number;
string name;
```

If the following function header is given:

```
void supplyInfo(float affordP, int nrP, string & nameP, float & amountP)
```

which of the options below is a correct calling statement of the function `supplyInfo`?

1. `supplyInfo(800, 4, "Beach Hotel", 789);`
2. `supplyInfo(800, 4, name, amount);`
3. `supplyInfo(affordP, nrP, nameP, amountP);`
4. `supplyInfo(afford, number, "Beach Hotel", 789);`
5. None of the above options is a correct calling statement.

QUESTION 3

Suppose the following declarations appear in a C++ program:

```
int age, days;
char grade;
float salary;
```

Suppose the following calling statement appears in the program:

```
inputInfo(35, 'B', salary, days);
```

Which of the options below is a correct function header of the function `inputInfo`?

1. `void inputInfo(int ageP, char gradeP, float & salaryP, int & daysP)`
2. `void inputInfo(35, 'B', float salaryP, int daysP)`
3. `void inputInfo(int & age, char & grade, float & salary, int & days)`
4. `void inputInfo(35, 'B', salary, days)`
5. All of the above options are correct function headers.

Questions 4 and 5 are based on the following C++ program.

```
#include <iostream>
#include <string>
using namespace std;

int main( )
{
  string favour;
  int tvHours, favourHours;

  cout << "How many hours per week do you watch sport on TV, " << endl
       << "what is your favourite sport, how many hours for that? ";
  cin >> tvHours >> favour >> favourHours;

  if (tvHours > 10)
    if (favour == "soccer")
    {
      if (favourHours > tvHours / 2)
        cout << "Group A" << endl;
    }
    else
      if (favourHours > tvHours / 4)
        cout << "Group B" << endl;
      else
        cout << "Group C" << endl;
  else
    if (favour != "rugby")
      cout << "Group D" << endl;

  return 0;
```

```
}
```

QUESTION 4

Suppose the program above is executed with the following input:

```
11 soccer 5
```

Which of the options below gives the output of the program?

1. `Group A`
2. `Group B`
3. `Group C`
4. `Group D`
5. No output

QUESTION 5

Suppose the following input is given to the program above:

```
6 soccer 6
```

Which of the options below gives the output of the program?

1. `Group A`
2. `Group B`
3. `Group C`
4. `Group D`
5. No output

QUESTION 6

Suppose in a C++ program values are assigned to a `string` variable `name` and to a `float` variable `amount`. Then a `while` loop is entered. The loop has to be executed as long as

- `name` is not equal to `"NOBODY"` and
- `amount` is not less than 10.00.

Which of the options below gives a correct condition for the loop?

1. `((name != "NOBODY") && (amount >= 10.00))`
2. `((name != "NOBODY") || (amount >= 10.00))`
3. `(!(name == "NOBODY") && (amount < 10.00))`
4. `(!((name != "NOBODY") && (amount >= 10.00)))`
5. None of the options above is a correct condition.

QUESTION 7

Suppose we want to assign the value `true` to a `bool` variable `success` if

- the value of the `int` variable `grade` is 10 or higher and

- the value of the `char` variable `sym` is `'A'` or `'B'` or `'C'`.

Which of the options below gives a correct assignment statement?

1. `success = (grade >= 10) || sym == 'A' || sym == 'B' || sym == 'C';`
2. `success = (grade >= 10) || (sym == 'A' || 'B' || 'C');`
3. `success = (grade >= 10) && (sym == 'A' || sym == 'B' || sym == 'C');`
4. `success = (grade >= 10) && sym == 'A' && sym == 'B' && sym == 'C';`
5. None of the options above is a correct assignment statement.

Questions 8, 9 and 10 are based on the following C++ program. Note that the conventions as explained in the Study Guide are used in the variable diagrams.

```
1     #include <iostream>
2     #include <string>
3     using namespace std;

4     void supplyInfo(string destP, float & feeP, string & timeP)
5     {
6       if (destP != "Cape Town" && feeP != 40)
7         feeP *= 3;
8       else if (destP != "Johannesburg")
9       {
10        destP = "Funny";
11        feeP = 1;
12      }

13      timeP = "17:15";
14      if (destP != "Funny")
15        timeP = "10:23";
16      else
17        timeP = "Fun";
18    }

19    float amountDue(int nrA, int nrC, float feeP)
20    {
21      float total;

22      total = nrA * feeP;
23      feeP *= 0.80;
24      total += nrC * feeP;

25      return total;
26    }

27    int main( )
28    {
29      string destination, time;
30      int nrAdult, nrChild;
31      float fee, amount;

32      destination = "P.E.";
```

```
33      fee = 40;
34      supplyInfo(destination, fee, time);
34      nrAdult = 4;
35      nrChild = 5;
36      amount = amountDue(nrAdult, nrChild, fee);
37      cout << "Time: " << time << "   Due: R" << amount << endl;

38      return 0;
39    }
```

QUESTION 8

Which of the options below correctly reflects the situation after Line 33 has been executed?

Option 1	Line 33

destination	fee	time
"P.E."	40	?

destP	feeP	timeP
"P.E."	40	?

nrAdult	nrChild	amount
?	?	?

Option 2	Line 33

destination | [destP]	fee | [feeP]	time | [timeP]
"P.E."	40	?

[nrAdult]	[nrChild]	[amount]
?	?	?

Option 3	Line 33

[destination] | destP	[fee] | feeP	[time] | timeP
"P.E."	40	?

[nrAdult]	[nrChild]	[amount]
?	?	?

Option 4	Line 33

destination	fee	time
"P.E."	40	?

nrAdult	nrChild	amount
?	?	?

Option 5	Line 33	None of the above options is correct.

QUESTION 9

Which of the options below correctly reflects the situation after Line 24 has been executed?

Option	Line
1	24

[destination]	[fee] | feeP	[time]
"Funny"	0.8	"Fun"

[nrAdult] | nrA	[nrChild] | nrC	[amount] | total
4	5	8

Option	Line
2	24

[destination]	[fee]	[time]
"P.E."	1	"Fun"

[nrAdult]	[nrChild]	[amount]
4	5	?

nrA	nrC	feeP	total
4	5	0.8	8

Option	Line
3	24

[destination]	[fee] | feeP	[time]
"P.E."	96	"10:23"

[nrAdult] | nrA	[nrChild] | nrC	[amount] | total
4	5	960

Option	Line
4	24

[destination]	[fee]	[time]
"P.E."	120	"10:23"

[nrAdult]	[nrChild]	[amount]
4	5	?

nrA	nrC	feeP	total
4	5	96	960

Option	Line	
5	24	None of the above options is correct.

QUESTION 10

Which of the options below correctly reflects the situation after Line 37 has been executed?

Option 1	Line 37

destination	fee	time
"P.E."	40	"10:23"

nrAdult	nrChild	amount
4	5	960

Option 2	Line 37

destination	fee	time
"P.E."	96	"10:23"

nrAdult	nrChild	amount
4	5	960

Option 3	Line 37

destination	fee	time
"Funny"	1	"Fun"

nrAdult	nrChild	amount
4	5	8

Option 4	Line 37

destination	fee	time
"Funny"	0.8	"Fun"

nrAdult	nrChild	amount
4	5	8

Option 5	Line 37	None of the above options is correct.

SECTION B (80 MARKS)

Please answer this section in your answer book.

QUESTION 1 **[12]**

A bookshop gives discount to customers as follows:

- Students get 10% discount,
- book dealers get 12% discount and
- pensioners get 15% discount.
- All other customers get 10% discount only if their total purchases are more than R200.

Write down ONLY the necessary C++ statements to calculate and display the final amount that is due, after discount.

Do NOT write a complete program. Use the following variables:

```
float amount;          // the amount due before discount
char customerType;     // the type of customer: 'S' (student) or
                       //  'D' (dealer) or 'P' (pensioner) or 'O'(other)
```

Assume that values have been assigned to `amount` and `customerType` already.

You may also need the following variables:

```
float discount, finalAmount;
```

QUESTION 2 **[10]**

A bank offers a fixed composite interest rate of 12% per year on money that is invested. Thus if you invest R100, it will be worth R112 at the end of the first year; and then R112 + (12% of R112) at the end of the second year, etc. *Complete the C++ program below* to determine the number of years needed for an initial investment of R1000 to become worth more than R3000. Use an appropriate loop structure.

Do not introduce any additional variables. Write down ONLY the missing statements.

```
#include <iostream>
using namespace std;

int main( )
{
  const float START = 1000;          // the amount that is invested
  const float PERC = 0.12;           // the interest rate
  float amount;                      // worth of investment
  int years;                         // number of years


// YOUR STATEMENTS SHOULD COME IN HERE

  cout << "After " << years << " years the investment "
       << "is worth more than R3000" << endl;

  return 0;
```

QUESTION 3 **[12]**

In this question you have to write a complete function.

Suppose marks have been given to 30 students and that the marks are stored in an `int` array called `marks`. You have to write a `void` function, called `findFailAndDistinct` to determine the number of students who failed (thus got a mark of less than 50) and the number of students who got a distinction (thus a mark of at least 75). These two values should be returned to the main function.

Assume the following:

- a declaration of a global constant:
  ```
  const int NUM_MARKS = 30;      // number of marks
  ```
- two declaration statements in the main function:
  ```
  int marks[NUM_MARKS];          // list of marks
  int nrFail, nrDistinct;
  ```
- values have been assigned already to all the elements of the array
- the function is called in the main function as follows:
  ```
  findFailAndDistinct(marks, nrFail, nrDistinct);
  ```

Write down ONLY the complete function `findFailAndDistinct`.

QUESTION 4 **[16]**

(a) Declare an integer constant `DIM1` equal to 3 and an integer constant `DIM2` equal to 5. (2)

(b) Declare two `int` two-dimensional arrays, namely
- `arrayA` with `DIM1` rows and `DIM2` columns, and
- `arrayB` with `DIM2` rows and `DIM1` columns. (4)

(c) Assume that values have been assigned to all the elements of `arrayA` and `arrayB`. Also assume that an `int` variable `funnySum` has been declared and intialised to 0. Use nested `for` loops and write down the necessary C++ instructions to calculate `funnySum` as follows: each row of `arrayA` is compared element by element to the corresponding column of `arrayB` and the value -1, 0, or 1 is added to `funnySum` according to whether the element of `arrayA` is less than, equal to, or greater than the element of `arrayB`. For example, if the given arrays `arrayA` and `arrayB` are:

```
8     12    5     10    3          6     6     5
6     7     4     9     16   and   6     7     9
3     2     20    22    15         20    4     4
                                   3     3     6
                                   22    4     5
```

respectively, then `funnySum` should be equal to

1+1-1+1-1 +0+0+0+1+1 -1-1+1+1+1 = 4.

Display the value of `funnySum`. Do NOT write a complete program or any functions. Write down ONLY the required statements. (10)

QUESTION 5 **[14]**

The Post Office keeps record of all parcels handled by them. They keep the following information for each parcel:

- sender (a string, for example `"Anna Every"`)
- receiver (a string, for example `"Johnny Allbody"`)
- weight (a floating point number, for example 2.25)
- postage (a floating point number, for example 23.50)

(a) Write down the declarations for a `struct` for storing the information of one parcel. Give the name `Parcel` to the `struct`. (5)

(b) Assume that an array

```
Parcel todaysParcels[50]
```

has been declared and that the information for 50 parcels has been stored in the array. The program fragment below determines and displays the average weight and the highest postage of all the parcels handled by the Post Office that day. Now write down ONLY the necessary C++ instructions for line numbers 4, 6, 7 and 9 to complete this program fragment. Write down only the line number and the instruction that should appear next to the line number. (9)

```
1.  float totalWeight, averageWeight, maxPostage;
2.  totalWeight = 0;
3.  maxPostage = 0;

4.  for (              ) //examine all parcels
5.  {
6.      // determine total weight of today's parcels
7.      // determine maximum postage of today's parcels
8.  }

9.  //  determine average weight of today's parcels;
```

```
10.  cout.setf(ios::fixed);
11.  cout.precision(2);
     cout << "The average weight of today's parcels is "
          << averageWeight << endl;
     cout << "The maximum postage for a parcel handled today is "
          << maxPostage << endl;
```

Do NOT write a complete program. Write down ONLY the required statements.

QUESTION 6 **[16]**

In both parts of this question you have to write the body of a function. **In both cases the function header looks as follows**:

```
string changedSentence(string senP)
```

Hint: Below the question we list a number of `string` member functions that you may need.

(a) The function receives a string of characters, indicated by `senP` in the function header. The function has to replace all occurrences of the string `he` with `Theo` and return the changed string to the main function.

Example: If the string

```
When he saw the hen, then and there he heard the noise.
```

is given, the string

```
WTheon Theo saw tTheo Theon, tTheon and tTheore Theo Theoard tTheo noise.
```

should be returned to the main function. You should write the body of the function ONLY. (8)

(b) The function receives a string of characters, indicated by `senP` in the function header. The function has to replace all occurrences of the ***stand-alone*** string `he` with `Theo` and return the changed string to the main function. You may assume that `senP` will not start or end with `he`.

Example: If the string

```
When he saw the hen, then and there he heard the noise.
```

is given, the string

```
When Theo saw the hen, then and there Theo heard the noise.
```

should be returned to the main function. You should write the body of the function ONLY. (8)

TURN OVER FOR SOME STRING MEMBER FUNCTIONS

A number of `string` member functions to help you

```
StringObject.size( )

StringObject.substr(startPos, length)

StringObject.find(substring)
StringObject.find(substring, startPos)

StringObject.insert(insertPos, substring);

StringObject.erase(startPos, length);

StringObject.replace(startPos, length, substring);
```

where

`startPos`, `length` and `insertPos` are of type `int`, and
`substring` is of type `string`.

CO MEMORANDUM

SECTION A

Question 1: 3
Question 2: 2
Question 3: 1
Question 4: 5
Question 5: 4
Question 6: 1
Question 7: 3
Question 8: 4
Question 9: 2
Question 10: 5

SECTION B

QUESTION 1 12 marks

```
// switch:
  switch(customerType)                                          1
  {
    case 'S': discount = 0.10; break;                           2
    case 'D': discount = 0.12; break;                           2
    case 'P': discount = 0.15; break;                           2
    case 'O':
      if (amount > 200)
        discount = 0.10;
      else
        discount = 0;                                           3
  }
  finalAmount = amount * (1.0 - discount);  // many other ways
                                                                1
  cout << finalAmount << endl;                                  1
```

OR

```
// nested if:
  if (customerType == 'S')
    discount = 0.10;                                            2
  else if (customerType == 'D')
    discount = 0.12;                                            2
  else if (customerType == 'P')
    discount = 0.15;                                            2
  else if (amount > 200)  //if (customerType == 'O') redundant
    discount = 0.10;                                            2
  else
    discount = 0.0;                                             2
  finalAmount = amount * (1.0 - discount);  // many other ways
                                                                1
  cout << finalAmount << endl;                                  1
```

QUESTION 2 **10 marks** (other loop structures may be used)

```
amount = START;                                                    1
years = 0;                                                         1
while (amount <= 3000)                                             3
{
  amount += PERC * amount;                                         3
  years++;                                                         2
}
```

QUESTION 3 **12 MARKS**

```
void findFailAndDistinct(int marksP[], int & nrFailP,
                                       int & nrDistinctP)    4
{
  nrFailP = 0;                                                     1
  nrDistinctP = 0;                                                 1
  for (int i = 0; i < NUM_MARKS; i++)                              2
   if (marksP[i] < 50)
     nrFailP++;                                                    2
   else if (marksP[i] >= 75)
     nrDistinctP++;                                                2
}
```

QUESTION 4: **2 + 4 + 10 = 16 MARKS**

QUESTION 4(a) **2 marks**

```
const int DIM1 = 3;                                                1
const int DIM2 = 5;                                                1
```

QUESTION 4(b) **4 marks**

```
 int arrayA[DIM1][DIM2];                                           2
 int arrayB[DIM2][DIM1];                                           2
```

QUESTION 4(c) **10 marks**

```
for (int i = 0; i < DIM1; i++)                                     2
  for (int j = 0; j < DIM2; j++)                                   2
  {
    if (arrayA[i][j] < arrayB[j][i])                               2
      funnySum--;
    else if (arrayA[i][j] > arrayB[j][i])                          2
      funnySum++;                                                  1
                                                  (together with --)
   }
cout << endl << endl << funnySum << endl;                          1
```

QUESTION 5: 5 + 9 = 14 MARKS

QUESTION 5(a) 5 marks

```
struct Parcel                                                    1
{
  string sender;                                                 1
  string receiver;                                               1
  float weight;                                                  1
  float postage;                                                 1
};
```

QUESTION 5(b) 9 marks

```
4     for (int i = 0; i < 50; i++)                               2
6     totalWeight = totalWeight + todaysParcels[i].weight;
                                                                 2
7     if (todaysParcels[i].postage > maxPostage)
        maxPostage = todaysParcels[i].postage;                   3
9     averageWeight = totalWeight/50;                            2
```

QUESTION 6: 8 + 8 = 16 MARKS

QUESTION 6(a) 8 marks

```
{
  int pos;

  pos = senP.find("he");                                         1
  while (pos > -1)                                               1
  {
    senP.replace(pos, 2, "Theo");                                2
    pos = senP.find("he", pos + 4);  // or pos+2 or pos+3
                                     // NOT pos or pos+1
                                                                 3
  }

  return senP;                                                   1
}
```

QUESTION 6(b) 8 marks

```
{
  int pos;

  pos = senP.find(" he ");                                       2
  while (pos > -1)                                               1
  {
    senP.replace(pos, 4, " Theo ");                              2
    pos = senP.find(" he ", pos + 7);
                                  // or pos, or pos+1, …         2
  }

  return senP;                                                   1
}
```

SECTION A 20 MARKS (10 MULTIPLE CHOICE QUESTIONS, 2 marks each)

Please answer this section on the **mark-reading sheet** that you received (**not** in your answer book).
Choose one option for every question.

QUESTION 1

Suppose the following declarations appear in a C++ program:

```
int afford, nr;
bool selfCater;
string whereToStay;
```

If the following function header is given:

```
string accommodation(int affordP, int nrP, bool selfCaterP)
```

which of the options below is a correct calling statement of the function `accommodation`?

1. `whereToStay = accommodation(650, 5, true);`
2. `whereToStay = accommodation(int affordP, int nrP, bool selfCaterP);`
3. `accommodation(affordP, nrP, selfCaterP);`
4. `whereToStay = accommodation(600, 6, "false");`
5. None of the above options is a correct calling statement.

QUESTION 2

Suppose the following declarations appear in a C++ program:

```
string destination, timeLeave;
float amount;
```

If the following function header is given:

```
void supplyTrainInfo(string & destP, string & timeLeaveP, float & amountP)
```

which of the options below is a correct calling statement of the function `supplyTrainInfo`?

1. `supplyTrainInfo(destP, timeLeaveP, amountP);`
2. `supplyTrainInfo(string destP, string timeLeaveP, float amountP);`
3. `supplyTrainInfo("Durban", "12:25", 527.95);`
4. `supplyTrainInfo(destination, '13:30', amount);`
5. None of the above options is a correct calling statement.

QUESTION 3

Suppose the following declarations appear in a C++ program:

```
string name, birthday, telNr;
int age;
```

Suppose the following calling statement appears in the program:

```
findInfo("Frank", 13, birthday, telNr);
```

Which of the options below is a correct function header of the function `findInfo`?

1. `void findInfo(string & nameP, int & ageP, string & birth, string & nr)`
2. `void findInfo("Frank", 13, string & birthP, string & nrP)`
3. `void findInfo(string nameP, int ageP, string & birth, string & nr)`
4. `void findInfo("Frank", 13, birthday, telNr)`
5. All of the above options are correct function headers.

Questions 4 and 5 are based on the following C++ program.

```
#include <iostream>
using namespace std;

int main( )
{
  int nrCoffee, timeFirst, timeLast;

  cout << "Number of cups of coffee that you drink per day, " << endl
       << "time of first cup and time of last cup (hours 0 to 24): ";
  cin >> nrCoffee >> timeFirst >> timeLast;

  if (nrCoffee < 8)
  {
    if (timeFirst < 7)
    {
      if (timeLast > 21)
        cout << "Group A" << endl;
    }
    else
      if ((timeFirst < 10) and (timeLast > 20))
        cout << "Group B" << endl;
  }
  else
    if (timeFirst > 5)
      cout << "Group C" << endl;
    else
      if ((timeLast - timeFirst) < 10)
        cout << "Group D" << endl;

  return 0;
```

```
}
```

QUESTION 4

Suppose the program above is executed with the following input:

```
12 5 14
```

Which of the options below gives the output of the program?

1. `Group A`
2. `Group B`
3. `Group C`
4. `Group D`
5. No output

QUESTION 5

Suppose the program above is executed with the following input:

```
6 10 22
```

Which of the options below gives the output of the program?

1. `Group A`
2. `Group B`
3. `Group C`
4. `Group D`
5. No output

QUESTION 6

Suppose in a C++ program values are assigned to an `int` variable `total` and to a `string` variable `name`. Then a `while` loop is entered. The loop has to be executed as long as

- `name` is not equal to `"Patience"` and
- `total` is less than 14.

Which of the options below gives a correct condition for the loop?

1. `((total < 14) || !(name == "Patience"))`
2. `(!(total > 14) && (name != "Patience"))`
3. `((total < 14) || (name != "Patience"))`
4. `((total < 14) && (name != "Patience"))`
5. None of the options above is a correct condition.

.

QUESTION 7

Suppose we want to assign the value `true` to a `bool` variable `safe` if

- either the value of the `int` variable `dogs` is greater than 0 and the value of the `char` variable `lock` is `'y'`, or
- the value of the `char` variable `alarm` is `'A'` or `'B'`.

Which of the options below gives a correct assignment statement?

1. `safe = (dogs > 0 || lock == 'y') && (alarm == 'A' || alarm == 'B');`
2. `safe = (dogs > 0 && lock == 'y') || (alarm == 'A' || 'B');`
3. `safe = (dogs > 0 && lock == 'y') || (alarm == 'A' || alarm == 'B');`
4. `safe = (dogs > 0 && lock == 'y') && (alarm == 'A' || alarm == 'B');`
5. None of the options above is a correct assignment statement.

Questions 8, 9 and 10 are based on the following C++ program and the input that is given below. Note that the conventions as explained in the Study Guide are used in the variable diagrams.

```
1     #include <iostream>
2     using namespace std;

3     void supplyTravelInfo(int & nrPasP, int & distP)
4     {
5       char dest;

6       cout << endl << "Destination and nr of passengers: ";
7       cin >> dest >> nrPasP;
8       switch (dest)
9       {
10        case 'A':  distP = 20;  break;
11        case 'B':  distP = 8;  break;
12        default:   distP = 10;
13      }
14      cout << endl;
15    }

16    float amountDue(int nrPasP, int distP, float feeP)
17    {
18      if (nrPasP > 6)
19        if (nrPasP > 10)
20          feeP *= 0.5;
21        else
22          feeP *= 0.6;
23      if (distP < 10)
24        feeP *= 1.5;
25      cout << "New fee has been calculated" << endl;
26      return (feeP * nrPasP);
27    }

28    int main( )
29    {
30      int nrPas, dist;
31      float basicFee, amount;

32      supplyTravelInfo(nrPas, dist);
```

```
33      basicFee = 10;
34      amount = amountDue(nrPas, dist, basicFee);
35      cout << "Amount due:  R" << amount << endl;

36      return 0;
37    }
```

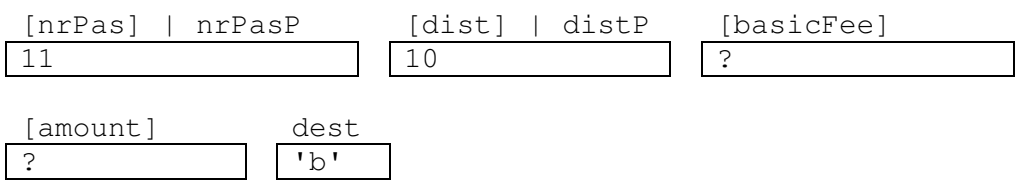
Suppose the program is executed with the following input:

```
      b     11
```

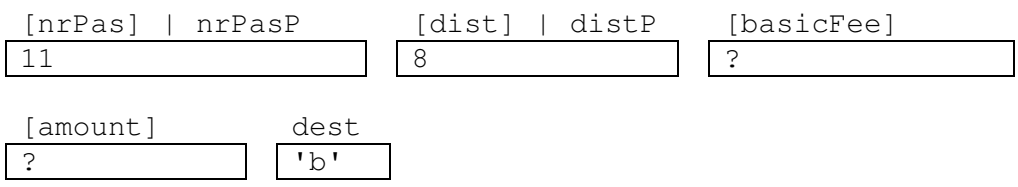
QUESTION 8

Which of the options below correctly reflects the situation after Line 14 has been executed?

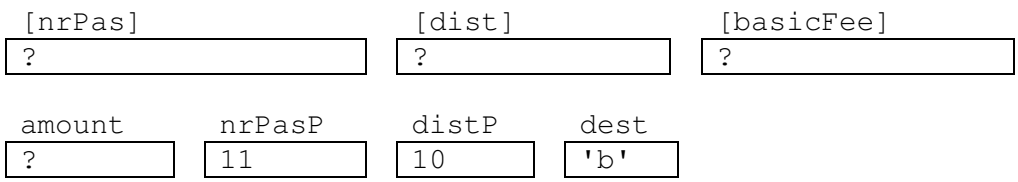
Option	Line
1	14



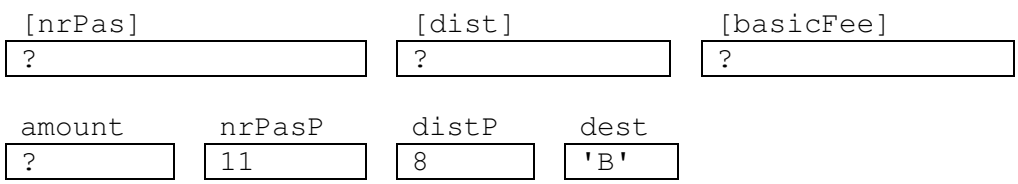
Option	Line
2	14



Option	Line
3	14



Option	Line
4	14



Option	Line	
5	14	None of the above options is correct.

QUESTION 9

Which of the options below correctly reflects the situation after Line 25 has been executed?

Option	Line
1	25

[nrPas] | nrPasP	[dist] | distP	[basicFee] | feeP
11	10	10

[amount]
?

Option	Line
2	25

[nrPas]	[dist]	[basicFee]
11	8	10

amount	nrPasP	distP	feeP
?	11	8	7.5

Option	Line
3	25

[nrPas]	[dist]	[basicFee]
11	10	10

amount	nrPasP	distP	feeP
?	11	10	5

Option	Line
4	25

[nrPas]	[dist]	[basicFee]
11	8	7.5

amount	nrPasP	distP	feeP
?	11	8	7.5

Option	Line	
5	25	None of the above options is correct.

QUESTION 10

Which of the options below correctly reflects the situation after Line 35 has been executed?

Option	Line
1	35

nrPas	dist	basicFee	amount
11	8	10	60

Option	Line	nrPas	dist	basicFee	amount
2	35	11	10	10	55

Option	Line	nrPas	dist	basicFee	amount
3	35	11	8	5	60

Option	Line	nrPas	dist	basicFee	amount
4	35	11	10	5	55

Option	Line	
5	35	None of the above options is correct.

SECTION B (80 MARKS)

Please answer this section in your answer book.

QUESTION 1 [12]

A cell phone company offers its customers a choice of three pricing options: Regular, Premium or Ultra. Calls are charged under these options as follows:

- Regular: R10 for the first 3 (or less) minutes of a call, and then R1 per minute.
- Premium: R20 for the first 10 (or less) minutes of a call, and then R1.50 per minute.
- Ultra: R45 flat rate per call for all calls not longer than 30 minutes. If a call is longer than 30 minutes, the rate is R2 per minute for the full duration of the call.

Write down ONLY the necessary C++ statements to calculate the cost and then display the cost and duration of a specific phone call.

Do NOT write a complete program. Use the following variables:

```
char option;     // 'R' (regular) or 'P'(premium) or 'U' (ultra)
int minutes;     // duration of the specific call
```

Assume that values have been assigned to `option` and `minutes` already.

You may also need the following variable:

```
float cost;
```

QUESTION 2 [10]

In an effort to encourage his son to stop smoking, Peter offers to give him R1 on the first day without smoking, then (an additional) R2 on the second day, R3 on the third day, and so on. Thus on the third day, the boy will have R1+R2+R3 = R6. *Complete the C++ program below* to determine the number of smoke-free days before the boy has at least R1000. Use an appropriate loop structure.

Do not introduce any additional variables. Write down ONLY the missing statements.

```
#include <iostream>
using namespace std;

int main( )
{
  float amount;
  int days;                          // number of smoke-free days

// YOUR STATEMENTS SHOULD COME IN HERE

  cout << "After " << days << " days he has at least R1000" << endl;

  return 0;
}
```

QUESTION 3 **[10]**

In this question you have to write a complete function.

Suppose the rainfall figures for the twelve months of 2008 are stored in a `float` array called `rain`. You have to write a function called `highestMonth` to determine the month (as an integer) which has the highest rainfall. If, for example, the highest rainfall occurred during March, the function has to return the number 3 to the main function.

Assume the following:

- a declaration of a global constant:
  ```
  const int NUM_MONTHS = 12;      // number of months
  ```
- two declaration statements in the main function:
  ```
  float rain[NUM_MONTHS];         // rainfall figures
  int month;
  ```
- values have been assigned already to all the elements of the array and the values are distinct (i.e. no value occurs more than once)
- the function is called in the main function as follows:
  ```
  month = highestMonth(rain);
  ```

Write down ONLY the complete function `highestMonth`.

QUESTION 4 **[18]**

Professor Knowal uses a two-dimensional array `marks` to store the test results of his first-year Computer Science students. There are 30 students in the class and the students write 5 tests during the course of the year.

(a) Declare an integer constant `NR_STUD` for the number of students and an integer constant `NR_TESTS` for the number of tests. (4)

(b) Declare a two-dimensional array `marks` with `NR_STUD` rows and `NR_TESTS` columns. (2)

(c) Assume that the following declarations are given:

```
int total, highest;
float average;
```

Further assume that values have been given to all the elements of the array `marks`. Use nested `for` loops and write down the necessary C++ instructions to determine and display the highest mark for each test as well as the average for each test.

Do NOT write a complete program or any functions. Write down ONLY the required statements. (12)

QUESTION 5 **[13]**

Cash registers use barcode readers to determine the price of an item. The following information is associated with each barcode:

- barcode itself (a string to store 9 digits, for example `123456789`)
- price (a floating point number, for example 23.50)
- description (a string, for example `"Magazine Fair Lady"`)

(a) Write down the declarations for a `struct` for storing the information associated with one barcode. Give the name `Barcode` to the `struct`. (4)

(b) Assume that an array

```
Barcode barcodeInfo[50]
```

has been declared and that the information for 50 barcodes has been stored in the array. The program fragment below does the following:

- Enter the barcode of an item that a customer wants to purchase and the number of these items that the customer wants to buy,
- search through the array `barcodeInfo` to find the price and description of the item with this barcode, and
- display the description, number of items and the amount due.

Now write down ONLY the necessary C++ instructions for line numbers 10, 12, 16 and 17 to complete the program fragment . Write down only the line number and the instruction that should appear next to the line number. (9)

```
1.   float totalDue;
2.   int nrOfItems;
3.   string barcodeForItem;
4.   int b;

5.   cout << "Enter barcode: ";
6.   cin >> barcodeForItem;
7.   cout << "Enter number of items: ";
8.   cin >> nrOfItems;

9.   b = 0;
     // examine all barcodes in barcodeInfo and compare with
     // barcodeForItem
10.  while ((b < 50) && ( ______ ))
11.    b++;
     //test if barcodeForItem has been found
```

```
12. if ( ____ )
13. {
14.   cout.setf(ios::fixed);
15.   cout.precision(2);
16.   // determine amount due
17.   // display description, number of items and amount due
18. }
19. else
20.   cout << "error";
```

Do NOT write a complete program. Write down ONLY the required statements.

QUESTION 6 **[17]**

In this question you have to write the body of a function. The function header is given. Hint: Below the question we list a number of `string` member functions that you may need.

The function header looks as follows:

```
string changedSentence(string senP)
```

The function receives a string of characters, indicated by `senP` in the function header. The function has to

- mark all occurrences of `s` in the given string, `senP`, by inserting the character `*` in front of `s`,
- replace all occurrences of `z` by `s`, and
- return the changed string to the main function.

Example: If the string

```
Susan realized that the organization's size was small.
```

is given, the string

```
Su*san realised that the organisation'*s *sise wa*s *small.
```

should be returned to the main function. You should write the body of the function ONLY.

TURN OVER FOR SOME STRING MEMBER FUNCTIONS

A number of `string` member functions to help you

```
StringObject.size( )

StringObject.substr(startPos, length)

StringObject.find(substring)
StringObject.find(substring, startPos)

StringObject.insert(insertPos, substring);

StringObject.erase(startPos, length);

StringObject.replace(startPos, length, substring);
```

where

`startPos`, `length` and `insertPos` are of type `int`, and
`substring` is of type `string`.

COS1 MEMORANDUM

SECTION A

Question 1: 1
Question 2: 5
Question 3: 3
Question 4: 4
Question 5: 5
Question 6: 4
Question 7: 3
Question 8: 1
Question 9: 3
Question 10: 2

SECTION B

QUESTION 1 12 marks

```
// switch:

  switch(option)                                                    1
  {
    case 'R':
      cost = 10;
      if (minutes > 3)
        cost += (minutes - 3);
      break;                                                        3
    case 'P':
      cost = 20;
      if (minutes > 10)
        cost += (minutes - 10) * 1.50;
      break;                                                        3
    case 'U':
      cost = 45;
      if (minutes > 30)
        cost = minutes * 2;
      break;                                                        3
  }
  cout << "Duration: " << minutes << " minutes" << endl
       << "Cost: R" << cost << endl;                                2
```

OR

```
// nested if:

  if (option == 'R')
  {
    cost = 10;
    if (minutes > 3)
      cost += (minutes - 3);
  }                                                                 3
  else if (option == 'P')
  {
    cost = 20;
    if (minutes > 10)
      cost += (minutes - 10) * 1.50;
  }                                                                 4
  else                // if (option == 'U') redundant
  {
    cost = 45;
    if (minutes > 30)
      cost = minutes * 2;
  }                                                                 3

  cout << "Duration: " << minutes << " minutes" << endl
       << "Cost: R" << cost << endl;                                2
```

QUESTION 2 **10 marks** (other loop structures may be used)

```
amount = 0;                                                    1
days = 0;                                                      1
while (amount < 1000)                                          3
{
  days++;                                                      2
  amount += days;                                              3
)
```

QUESTION 3 **10 marks**

```
int highestMonth(float rainP[])                                3
{
  int m;
  m = 0;                                                       1
  for (int i = 1; i < NUM_MONTHS; i++)                         2
   if (rainP[i] > rainP[m])
     m = i;                                                    2
  return m + 1;                                                2
}
```

QUESTION 4: **4 + 2 + 12 = 18 MARKS**

QUESTION 4(a) **4 marks**

```
const int NR_STUD = 30;                                        2
const int NR_TESTS = 5;                                        2
```

QUESTION 4(b) **2 marks**

```
 int marks[NR_STUD][NR_TESTS];                                 2
```

QUESTION 4(c) **12 marks**

```
for (int i = 0; i < NR_TESTS; i++)                             2
  {
    total = 0;                                                 1
    highest = 0;                                               1
    for (int j = 0; j < NR_STUD; j++)                          2
    {
      total += marks[j][i];                                    1
      if (marks[j][i] > highest)
        highest = marks[j][i];                                 2
     }
     average = float(total) / NR_STUD;                         1
     cout << "Average mark and highest mark for test "
          << i+1 << " are "  <<  average << " and "
          << highest << endl;                                  2
   }
```

QUESTION 5: **4 + 9 = 13 MARKS**

QUESTION 5(a) **4 marks**

```
 struct Barcode                                          1
 {
   string code;                                          1
   float price;                                          1
   string description;                                   1
};
```

QUESTION 5(b) **9 marks**

```
10 while ((b < 50) && (barcodeForItem != barcodeInfo[b].code))
                                                         2
12 if (b < 50)                                           2
16 totalDue = nrOfItems * barcodeInfo[b].price;          2
17 cout << barcodeInfo[b].description << "  " << nrOfItems
        << "   R" << totalDue << endl;                   3
```

QUESTION 6 **17 marks**

```
{
  int pos;

  pos = senP.find("s");                                  2
  while (pos > -1)                                       1
  {
    senP.insert(pos, "*");                               2
    pos = senP.find("s", pos + 2);  // NOTE +2           3
  }

  pos = senP.find("z");                                  2
  while (pos > -1)                                       1
  {
    senP.replace(pos, 1, "s");                           2
    pos = senP.find("z", pos + 1);  // or pos            3
  }

  return senP;                                           1
}
```

SECTION A 20 MARKS (10 MULTIPLE CHOICE QUESTIONS, 2 marks each)

Please answer this section on the **mark-reading sheet** that you received (**not** in your answer book).
Choose one option for every question

QUESTION 1

Suppose the following declarations appear in the `main` function of a C++ program:

```
float callDuration, costPerSecond, callAmount;
```

If the following function header is given

```
float calcCost(float callDurationP, float costPerSecondP)
```

which of the options below is a correct calling statement of the function `calcCost` in the `main` function?

1 `callAmount = calcCost(float callDuration, float costPerSecond),`
2 `callAmount = calcCost(callDurationP, costPerSecondP);`
3 `calcCost(callDuration, costPerSecond);`
4 `calcCost(9.00, 0.25),`
5 None of the above is a correct calling sequence

QUESTION 2

Suppose the following declarations appear in the `main` function of a C++ program:

```
string ageRestriction,
int violenceRating, languageRating,
```

If the following function header is given

```
void classifyMovie(int violenceRatingP, int languageRatingP,
                   string & ageRestrictionP);
```

which of the options below is a correct calling statement of the function `classifyMovie` in the `main` function?

1 `classifyMovie(2,0,"ALL");`
2 `classifyMovie(violenceRatingP, languageRatingP,`
`ageRestrictionP);`
3 `classifyMovie(violenceRating, languageRating, "UNDER 18");`
4 `classifyMovie(violenceRating, languageRating, ageRestriction);`
5 None of the above is a correct calling sequence

QUESTION 3

Suppose the following declarations appear in the `main` function of a C++ program

```
string show, date;
int seatNo;
float price,
```

Suppose the following call statement appears in the main function:

```
bookTicket("Cinderella on Ice", "23/12/2011", seatNo, price);
```

Which of the options below is a correct function header of the function bookTicket in the main function?

```
1 void bookTicket(string & showP, string & dateP,
                  int seatNoP, float  priceP)
2 void bookTicket(string & showP, string & dateP,
                  int & seatNoP, float & priceP)
3 void bookTicket(string showP, string dateP,
                  int & seatNoP, float & priceP)
4 void bookTicket(showP, dateP, seatNoP, priceP)
```
5 All of the above options are correct function headers

QUESTION 4

Consider the following C++ code segment and answer the question below

```
int x = 25, y = 9, z = 17;
if (x > y)
{
  if (x < z)
    x = y,
}
else if (x > z)
       x = 0;
         else x = z;
```

What value will x have after the following code has been executed?

1 25
2 9
3 17
4 0
5 Undefined

QUESTION 5

Suppose the input value for alpha is 5 What is the value of alpha after the following C++ code has been executed?

```
cin >> alpha;
switch (alpha)
{
      case 1.
      case 2: alpha = alpha + 2,
              break;
      case 4: alpha++;
      case 5: alpha = 2 * alpha;
      case 6: alpha = alpha + 5;
              break;
      default: alpha--;
```

```
}
```

1 10
2 7
3 15
4 17
5 4

QUESTION 6

Consider the following C++ code segment

```
int calcResult(int iLimit)
{
int iIndex = 0;
int iResult = 0;
while (iIndex <= iLimit)
{
      iResult += iIndex,
      iIndex++;
}

return iResult,
}
```

What will be the output of the following statement executed in the `main` function?

```
cout << calcResult(3);
```

1 3
2 6
3 0
4 4
5 None of the above

QUESTION 7

What will be displayed on the screen by the following code?

```
int number = 3;
while (number == 3)
{
      cout << number << " + ",
      number++,
}
```

1 3
2 3 +
3 3 + 4
4 3 + 4 + 5 + 6 + (i e the next value will be + 7 followed by + 8 and so forth)
5 3 + 3 + 3 + 3 + (i e the next value will be + 3 followed by + 3 and so forth)

QUESTION 8

The following segment of code checks whether a number is within a given range Note

- MIN and MAX are both `const int` variables, where `MAX > MIN`.
- `iNum` is an `int` variable,
- `bValid` is a `bool` variable.

```
if ((iNum < MIN) || (iNum > MAX))
   bValid = false;
else
   bValid = true,
```

Which of the following pieces of code will give exactly the same result as the code above?

```
1 if ((iNum >= MIN) && (iNum <= MAX))
    bValid = true;
  else
    bValid = false;

2 if ((iNum > MIN) && (iNum <  MAX))
    bValid = true;
  else
    bValid = false;

3 if ((iNum > MIN) || (iNum <  MAX))
    bValid = false;
  else
    bValid = true;

4 if ((iNum <= MIN) || (iNum >=  MAX))
    bValid = false;
 else
   bValid = true;
```

5 None of the above

QUESTION 9

Stella wants to follow the guidelines given by the Garden Society of SA for affordable and waterwise gardening
Plant indigenous plants (represented by a Boolean variable `indigenous`), or edible plants (represented by a Boolean variable `edible`), and do not pay more than R30 00 for a plant A `float` variable `price` indicates the price of a plant The Boolean variable `plantOK` should receive the value `true` if a plant satisfies these conditions. Which of the following options will assign the correct value to `plantOK`?

```
1  plantOK = (indigenous || edible) && ( price < 30.00);

2  plantOK = (indigenous || edible) && ( price <= 30.00);

3  plantOK = (indigenous && edible) && ( price <= 30.00);

4  plantOK = (indigenous && edible) && ( price < 30.00);

5  plantOK = (indigenous || edible) || ( price < 30.00);
```

QUESTION 10

Which of the following statements is not a valid C++ statement?

1 `x = y = z = 14;`
2 `x =* 14;`
3 `if (x == 14) x = 0;`
4 `x = (y * (x + (z - 4)));`
5 All of the above statements are valid C++ statements

SECTION B 70 MARKS

QUESTION 1 **[4]**

(a) Assume that a, b, and c have been declared as integers and have been initialized Explain in words what the purpose of the following segment of code is [2]

```
a = b;
b = c;
c = a;
```

(b) Explain the purpose of the following segment of code [2]

```
int a[5] = {1,2,3,4,5};
const N = 5;
c = 0;
for (int i = 0; i < N; i++)
  c += a[i];
int d = c/N;
```

QUESTION 2 **[6]**

Question 2 is based on the following program

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4 const int CD = 50;
5 const int DISCOUNT = 100;
6 const int FEE = 800;
7 int updateDetails(string subjectP, string & codeP, char indP);
8 {
9     int total = FEE;
10    if (subjectP == "PROGRAMMING" && codeP == "COS111U")
11    {
12        codeP = "COS1511",
13        if (indP == 'n')
14           total -= DISCOUNT;
15        else{
16           total += CD,
17    }
18    else if (codeP == "COS112V")
19    {
20        codeP = "COS1512";
```

```
21          subjectP = "PROGRAMMING 2";
22      }
23      else if (codeP == "COS101S")
24      {
25          total += CD + 50;
26          codeP = "COS1501";
27      }
28      indP = 'y';
29      return total,
30  }
31  int main()
32  {
33      string subject = "PROGRAMMING", code,
34      char ind = 'n';
35      code = "COS112V";
36      int cost,
37      cost = updateDetails(subject, code, ind);
38      cout << "R" << cost << " owed for subject " << subject << endl;
39      return 0,
40  }
```

Below is part of the variable diagram for the above program We only show lines that change the value of at least one variable in the variable diagrams Please answer the questions that follow the variable diagram

Line nr	[subject]	subjectP	[code] codeP	[ind]	indP	total
line 37->7	"PROGRAMMING"	"PROGRAMMING"	"COS112V"	'n'	'n'	-
line 9	"PROGRAMMING"	"PROGRAMMING"	"COS112V"	'n'	'n'	800
line ?						
line ?						
line 28						

(a) Complete the two missing line numbers above as well as the values of the variables after these lines have been executed in your answer books, as follows
If you think for example, that the two lines that will change the values of variables after line 9 has been executed, are lines 25 and 26, answer as follows in your answer book by supplying the line numbers and giving the correct values for the variables after each line has been executed respectively

line 25	"PROGRAMMING"	"PROGRAMMING"	"COS101S"	'n'	'n'	900
line 26	"PROGRAMMING"	"PROGRAMMING"	"COS1501"	'n'	'n'	900

[2]

(b) Complete the values of the variables after line 28 has been executed the same way as in question 2(a) [1]

(c) What is the value of the variable `subject` after line 37 has been executed? [1]

(d) What is the value of the variable `code` after line 37 has been executed? [1]

(e) What is the value of the variable `ind` after line 37 has been executed? [1]

QUESTION 3 **[13]**

The classes of the Princess Primary School are partaking in a parade Each class will wear a different colour costume, depending on the number of pupils in the class

- Classes with 11 to 20 pupils are wearing green costumes
- Classes with 21 to 30 pupils are wearing red costumes
- If a class has between 31 and 40 pupils, and all of them are boys, they will be wearing blue costumes If all of them are girls, they will be wearing pink costumes If the class consists of boys and girls, the whole class will be wearing white costumes
- Any class with less than 11 or more than 40 pupils will not participate in the parade

Write a **nested if** statement to assign a value to the colour of the costumes according to the number of pupils in the class and to display the colour or an appropriate message if the class may not participate

Do NOT write a complete program Use the following variables

```
string colour,  //can have the value of "blue", "red", "green", "white" or
                //"pink"
int number;     //number of pupils in class
char gender,    //can be 'm' for male, 'f' for female, or 'b' for a class with girls and boys
```

Assume that values are already assigned to variables `number` and `gender`

QUESTION 4 **[8]**

Complete the C++ program below, which is based on a guessing game The user enters two large numbers and then has to guess the sum of the two numbers The user is given a maximum of 5 guesses The program continually prompts the user for a guess until the user guesses correctly or until the maximum number of tries is reached Once the user guesses correctly the program outputs the statement "You Guessed Correctly" Use an appropriate loop structure

Do not add any additional variables Write down ONLY the missing statements

```
#include <iostream>
using namespace std,

int main()
{
  int num1, num2,
  int sum,                  // sum of num1 and num2;
  int userGuess;            // the number guessed by the user
  const int MAXGUESSES = 5,      // maximum number of guesses allowed
  int noOfGuesses ;         // the number of guesses by the user
  bool userCorrect,         // user has guessed correctly or not

  noOfGuesses = 0;
  userCorrect = false,

  cout << "Enter two large integer values " << endl;
```

```
   cin >> num1 >> num2;
   sum = num1 + num2,

    //YOUR STATEMENTS SHOULD COME IN HERE

   if ('userCorrect)
   {
      cout<<"Sorry better luck next time!" << endl;
      cout<<"The sum was " << sum;
   }
   return 0;
}
```

QUESTION 5 **[12]**

In this question you have to write a complete function

Peter is making a study of rainfall figures for last year The names of the 12 months of the year are kept in a one-dimensional `string` array `month`. The rainfall figure for each month is kept in a one-dimensional `float` array `rainfall` You have to write a `void` function `findLowest` that will do the following

- return the name of the month with the lowest rainfall,
- the function is not allowed to change any values in the two arrays `month` and `rainfall`,
- count how many months had a rainfall figure of 0 00 and return this value,
- display the months and the rainfall figures, for example if January has a rainfall figure of 23 10 and February has a rainfall figure of 10 23, it will display as follows

```
Month: January  Rainfall: 23 10
Month: February Rainfall: 10.23
```

etc

The following variables have already been defined in the `main` function

```
const int NUM = 12;              //number of months
string month [NUM],              //array containing the month,
                                  //e.g. "January", "February" etc
float rainfall [NUM];             //array containing the
                                  //corresponding rainfall figures
float lowest;                    //to hold the lowest rainfall figure
int count = 0;                   //to hold number of months with a
                                  //rainfall figure of 0.00
```

Assume the following

- values have been assigned already to all the elements of the arrays
- the function is called in the main program as follows

```
findLowest (month, rainfall, lowest, count);
```

Write down ONLY the complete function `findLowest`.

QUESTION 6 **[15]**

Consider the following declarations below

```
const int DEPARTMENT = 4;
const int STORE = 3;
```

(a) Declare a two-dimensional integer array called `inStaff` with `DEPARTMENT` number of rows and `STORE` number of columns [2]

(b) Suppose `inStaff` represents the number of staff that are employed in 3 stores and 4 departments Write down ONLY the necessary C++ instructions to determine the range of employees The range is the difference between the highest and lowest value stored in `inStaff` You MUST use nested `for` loops to iterate through the array

For example, if the given array for `inStaff` is

4	15	5
5	2	9
6	7	5
6	9	3

The highest value in the array is 15 and the lowest value is 2, hence
range = 15 – 2 = 13
Accordingly the output would be `Range of employees is 13`

Do NOT write a complete program or any functions Assume that the values stored in the array are unknown to you
Hint: Declare two `int` variables `highest` and `lowest` and assign the highest and lowest value in the array accordingly [13]

QUESTION 7 **[6]**

A local restaurant wants to automate its breakfast billing system

(a) Define a struct `MenuItemType` with the following components
- `menuItem` (for example `"Muffin"` of type `string`)
- `menuPrice` (for example `10.99` of type `float`) [3]

(b) Assuming the following declarations
```
MenuItemType customerOrder[3];
float amountDue = 0;
```

Complete the code segment below, in order to determine the total amount due by the customer and to output the bill for the customer Assume that the values have already been assigned to all the elements of `customerOrder`

```
for ( int i = 0; i < 3, i++)
{
      //YOUR STATEMENTS SHOULD COME IN HERE
}
cout << "Amount Due  ....R " << amountDue << endl,
```

A sample output would be

```
Bacon & Egg ..  R 24.99
Bran Muffin ... R 12 99
Milk Coffee ... R 2.99
Amount Due  ....R 40.97
```
[3]

QUESTION 8 **[6]**

In this question the `main` function is given You have to write a function `removeChar()` that is called from the `main` function with the following function header

```
string removeChar(string inStringP)
```

The function receives a string of characters, indicated by parameter `inStringP` in the function header The function has to do the following

- Find all the occurrences of "p" in a given string and erase them
- Return the changed string to the `main` function

Example If the string

```
Peter Piper picks a peck of pickled peppers
```

is given, the string

```
Peter Pier icks a eck of ickled eers
```

should be returned to the main function Write only the function `removeChar()`

```
#include <iostream>
#include <string>
using namespace std,

//Insert function removeChar here

main()
{
    string sentence,
    // Prompt user to enter a sentence
    cout << "Enter sentence :" << endl;
    // Read the sentence from the user
    getline (cin, sentence, '\n'),
    cout << "New sentence  "<< endl;
    cout << removeChar(sentence) << endl,,
   return 0,

}
```

Please turn the page for a number of string functions to help you

A number of `string` member functions to help you

```
StringObject.size( )

StringObject.substr(startPos, length)

StringObject.find(substring)
StringObject.find(substring, startPos)

StringObject.insert(insertPos, substring);

StringObject.erase(startPos, length);

StringObject.replace(startPos, length, substring);
```

where

`startPos`, `length` and `insertPos` are of type `int`, and
`substring` is of type `string`

COS1511/COS111U MEMORANDUM

NB: Please make a tick for each mark – it is required by the new assessment procedure, i.e. if the student obtains 7/10 for a question, there should be 7 √'s

These answers are guidelines. Please give marks for correct logic, even if answered different to the solutions. If needed, write a note to the moderator of how marks were given in this case.

Do not subtract marks for the same error twice in the same question. (especially question 3)

SECTION A **20 marks**

Question 1: 5
Question 2: 4
Question 3: 3
Question 4: 1
Question 5: 3
Question 6: 2
Question 7: 2
Question 8: 1
Question 9: 2
Question 10: 2

Each question 2 marks

SECTION B **70 marks**

QUESTION 1 **4 marks**

(a) Swaps `b` and `c`'s values. 2
If student gives a line by line explanation, e.g. "assigns `b`'s value to `a`, `c`'s value to `b`, etc", only 1 mark.

(b) Calculate the average of the elements in array `a` 2
If student gives a line by line explanation, only 1 mark.

QUESTION 2 **6 marks**

Note to markers: For questions 2(a) and (b) the mark is given for the highlighted field. For questions 2(c) – (e), if the student left out the single and double quotes, but the value is correct, please give the mark.

(a)

line 20	"PROGRAMMING"	"PROGRAMMING"	"COS1512"	'n'	'n'	800
line 21	"PROGRAMMING"	"PROGRAMMING 2"	"COS1512"	'n'	'n'	800

[2]

(b)

line 28	"PROGRAMMING"	"PROGRAMMING 2"	"COS1512"	'n'	'y'	800

[1]

(c) "PROGRAMMING" [1]
(d) "COS1512" [1]

(e) 'n' [1]

QUESTION 3 13 marks

Note to markers: Student must use a nested if loop, as it is stated in the question.

Students may have put the cout statement within each if or else in which case they will have 5 cout statements for the colour and one for the error message. Make sure { } pairs have been used in this case. If no { } pairs have been used, subtract 1 mark for incorrect logic. Only one mark for cout for fee and one mark for error message cout.
`if (number > 10 && number < 21) is` also correct. Same for other `if`'s

```
if (number >= 11 && number <= 20)                          [1]
       colour = "green";                                   [1]
else if (number >= 21 && number <= 30)                     [1]
        colour = "red";                                    [1]
     else if (number >= 31 && number <= 40)                [1]
            if (gender == 'm')                             [1]
               colour = "blue";                            [1]
            else if (gender == 'f')                        [1]
                   colour = "pink";                        [1]
                else colour = "white";                     [1]
if (number < 11 || number > 40)                            [1]
     cout << "Class will not participate" << endl;         [1]
else
     cout << "The colour is " << colour << endl;           [1]
```

QUESTION 4: 8 MARKS

Give marks for any correct `cout` statement asking the user to guess a number. The student could also have tested `userCorrect` inside the `while` loop instead of testing it as one of the conditions – give marks if logic is correct.
(Many students answered this question without using `userCorrect`. We gave marks for correct logic).

```
#include <iostream>
using namespace std;

int main()
{
      int num1, num2;
      int sum;                  //sum of num1 and num2;
      int userGuess;            // the number guessed by the user
      const int MAXGUESSES = 5;//maximum number of guesses allowed
      int noOfGuesses ;        // the number of guesses by the user
      bool userCorrect;        //user has guessed correctly or not

      noOfGuesses = 0;
```

```
      userCorrect = false;
      cout << "Enter two large integer values:" << endl;
      cin >> num1 >> num2;
      sum = num1 + num2;

      while (!userCorrect ✓&& noOfGuesses < MAXGUESSES✓)          2
              (OR while (!userCorrect ✓&& noOfGuesses < 5✓)
      {
       cout << "Guess the sum of " << num1 <<" + " << num2 <<
                                        " = ?"  << endl; ✓          1
          cin >> userGuess;    ✓                                   1
          if(userGuess == sum)       ✓{                           1
            cout<<"You Guessed Correctly" << endl; ✓               1
            userCorrect=true; ✓                                    1
          }
            noOfGuesses++;✓                                        1
      }

      if (!userCorrect)
      {
            cout<<"Sorry better luck next time!" <<endl;
            cout<<"The sum was "<<sum;
      }
      return 0;
}
```

QUESTION 5: 12 MARKS

Note to markers: The first two parameters must be `const` variables – deduct ½ mark for each one that does not have the `const`.
`const string monthP[12]` and `const float rainfallP[12]` is also correct.

```
void findLowest (const string monthP[], const float
        rainfallP[], float & lowestP, int & countP)    [4]
{
      lowestP = 1000.00; //(or any high value)          [1]
                         //(or lowest = rainfall[0];)
//We gave marks here for any initialisation
      for (int i = 0; i < NUM; i++)                     [1]
      {
         if (rainfallP[i] < lowestP)                    [1]
            lowestP = rainfallP[i];                     [1]
         if (rainfallP[i] == 0.00)                      [1]
            countP ++;                                  [1]

        cout << "Month: " << monthP[i] <<
          "Rainfall: " << rainfallP[i] << endl;        [2]
      }

}
```

QUESTION 6 15 marks

Note to markers: Give 1 marks for `int inStaff[4][3].`
We did not specify whether the array had to read row-wise or column-wise. Give marks for both.

(a)

```
int inStaff[DEPARTMENT][STORE]; ✓✓                                  2
```

(b)

```
int lowest = inStaff[0][0]; ✓ //or any high value                   1
int highest = inStaff[0][0]; ✓ //or any low value                   1
//we gave marks here for any initialisation of lowest and highest
int range;                                                          1
for(int i = 0; i < DEPARTMENT;i++){✓✓                               2
      for (int j = 0; j < STORE; j++)✓✓                             2
//we gave marks here if 4 and 3 were used i.s.o. DEPARTMENT and STORE
{
   if (inStaff[i][j] > highest) ✓                                   1
   {
      highest = inStaff[i][j]; ✓                                    1
   }
   if (inStaff[i][j] < lowest) ✓                                    1
   {
      lowest=inStaff[i][j]; ✓                                       1
   }
}
}

range = highest - lowest; ✓                                         1
cout<<" The range of employee is " << range; ✓                      1
```

QUESTION 7 6 marks

Note to markers: Do not deduct marks if " . . . " is not included in the `cout` statement

(a)

```
struct MenuItemType✓                                                1
{
 string menuItem; ✓                                                 1
 float menuPrice; ✓                                                 1
};
```

(b)

```
 for ( int i = 0; i < 3; i++)
{
      cout  <<customerOrder[i].menuItem✓                            1
            <<" ... R "<<customerOrder[i].menuPrice<<endl; ✓1
      amountDue = amountDue + customerOrder[i].menuPrice; ✓ 1

}
cout<<"Amount Due .....R " << amountDue <<endl;
```

QUESTION 8 6 marks

```
string removeChar(string inStringP)
{
   int pos;                                                  1

   // Find 'p' and erase it from the string
   pos = inStringP.find("p");                                1
   while (pos > -1)                                          1
   {
       inStringP.erase(pos, 1);                              1
       pos = inStringP.find("p", pos);  //NOT pos+1          1
   }
   return inString;                                          1
}
```

SECTION A 20 MARKS (10 MULTIPLE CHOICE QUESTIONS, 2 marks each)

Please answer this section on the **mark-reading sheet** that you received (**not** in your answer book).
Choose one option for every question

QUESTION 1

Suppose the following declarations appear in the `main` function of a C++ program:

```
float itemPrice, discount, totAmount;
int number;
```

If the following function header is given:

```
float calcPrice(float itemPriceP, int numberP, float discountP)
```

which of the options below is a correct calling statement of the function `calcPrice` in the `main` function?

```
1 calcPrice(itemPrice, number, discount);
2 calcPrice(29.95, 2, 0.10);
3 totAmount = calcPrice(39.50, 50, 0.12);
4 totAmount = calcPrice(float itemPriceP, int numberP, float discountP);
```

QUESTION 2

Suppose the following declarations appear in the `main` function of a C++ program:

```
string studentName;
int studentNumber;
float fees;
```

If the following function header is given:

```
void studentFees(int studentNumberP, string studentNameP,
                 float & feesP);
```

which of the options below is a correct calling statement of the function `studentFees` in the `main` function?

```
1 studentFees(346577, "Mr R.M. Seopa", 1520.00);
2 studentFees(int studentNumberP, string studentNameP, float & feesP);
3 studentFees(studentNumber, studentName, 1740.00);
4 studentFees(777543, "Ms T. Mokwana", fees);
```

QUESTION 3

Suppose the following declarations appear in the `main` function of a C++ program:

```
int stNumber;
float examMark, yearMark, finalMark;
```

Suppose the following calling statement appears in the `main` function:

```
updateTotal(67.50, 55.60, finalMark, stNumber);
```

Which of the options below is a correct function header of the function `updateTotal`?

```
1 void updateTotal(examMarkP, yearmarkP, finalMarkP, stNumberP)
2 void updateTotal(float examMarkP, float yearmarkP, float & finalMarkP,
                                                        int stNumberP)
3. void updateTotal (float & examMarkP, float & yearMark,
                                        float & finalMarkP, int stNumberP)
4 void updateTotal (float & examMarkP, float & yearMark,
                                          float finalMarkP, int stNumberP)
```

QUESTION 4

Consider the following C++ code segment and answer the question below:

```
int value1 = 20, value2 = 5;
int max;
if (value2 - value1 > 0)
   max = value1 + value2;
else if (value1 == value2)
        max = 2 * value1;
     else
        max = value1 + value2 * 2;
```

What value will `max` have after the code above has been executed?

1 50
2 40
3 30
4 25

QUESTION 5

Suppose the input value for x is 3. What is the value of x after the following C++ code has been executed?

```
int x;
cin >> x;
switch (x)
{
      case 1: x = x * 2;
              break;
      case 3: x =  x * 3;
      case 5: x -= 3;
      case 6: --x;
              break;
      default: x = x + 10;
}
```

1 5
2 6
3 9
4 15

QUESTION 6

Consider the following C++ code segment:

```
int calcFunny(int choiceP)
{
  int counter = 5;
  int funnySum = 35;
  while (counter > choiceP)
  {
      funnySum -= counter;
      counter --;
  }

return funnySum;
}
```

What will be the output of the following statement executed in the `main` function?

```
cout << calcFunny(2);
```

1 47
2 35
3 23
4 20

QUESTION 7

What will be displayed on the screen by the following code?

```
int number = 1;
int count = 0;
while (number != 10)
{
      cout << number << " ; ";
      count++;
}
```

1 There will not be any output
2 1 ;
3 1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 7 ; 8 ; 9 ;
4 1 ; 1 ; 1 ; 1 ; 1 ; . .(i e the next value will be 1 followed by ; and so forth .)

QUESTION 8

Consider the following code fragment. Note that `colour` is a `char` variable and `size` is an `int` variable. If at least one of the conditions is true, `choice` is set to `'y'`, else `choice` is set to `'n'`

```
if (colour == 'r' || size == 34 )
    choice = 'y';
else
    choice = 'n';
```

Which of the following pieces of code will give exactly the same result as the code above?

```
1  if (!(colour == 'r' || size == 34))
      choice = 'n';
   else
      choice = 'y';

2  if ('(colour == 'r' && size == 34 ))
      choice = 'y';
  else
      choice = 'n';

3. if (colour == 'r' && size == 34 )
      choice = 'y';
  else
      choice = 'n';

4. if (colour == 'r')
      if (size == 34)
         choice = 'y';
      else
         choice = 'n';
```

QUESTION 9

The *BargainBooks* shop has a sale. A book is on sale under the following conditions:

- There are at most 3 books left with the title and no books with this title have been sold in the last month, OR
- The book was published before 1980

An `int` variable `number` indicates the number of books left. An `int` variable `sold` indicates the number of books with this title sold in the last month. An `int` variable `date` indicates the published date of the book. The Boolean variable `sell` should receive the value `true` if a book satisfies these conditions. Which of the following options will assign the correct value to `sell`?

1. `sell = (number == 3 && sold == 0) && ( date < 1980);`

2 `sell = (number <= 3 && sold == 0) || ( date < 1980);`

3 `sell = (number <= 3 && sold == 0) && ( date < 1980);`

4 `sell = (number == 3 || sold == 0) || ( date < 1980);`

QUESTION 10

Which one of the following variable names is NOT a valid variable name in C++?

```
1 string invalidName;
2 int rT5_87G_hV2;
3 char right-colour;
4 string COS1511;
```

SECTION B 70 MARKS

QUESTION 1 **[4]**

In Questions 1(a) and (b) you have to write down what the purpose of the segment of code is. Look at the following example before answering the questions:

```
int a,b,c;
cin >> a >> b >> c;
cout << c + b + a;
```

The purpose of the above code segment is input three integer values and display their sum. Now answer questions 1(a) and (b) below.

(a)

Assume that `value1`, `value2` and `value3` have been declared as integers. Explain in words what the purpose of the following segment of code is: [2]

```
cin >> value3 >> value2 >> value1;
cout << value1 << value2 << value3;
```

(b)

Explain the purpose of the following segment of code: [2]

```
string colours[5] = {"red", "blue", "red", "green", "yellow"};
int count = 0;
for (int j = 0; j < 5; j++)
{
  if (colours[j] == "red")
      count ++;
}
```

QUESTION 2 **[11]**

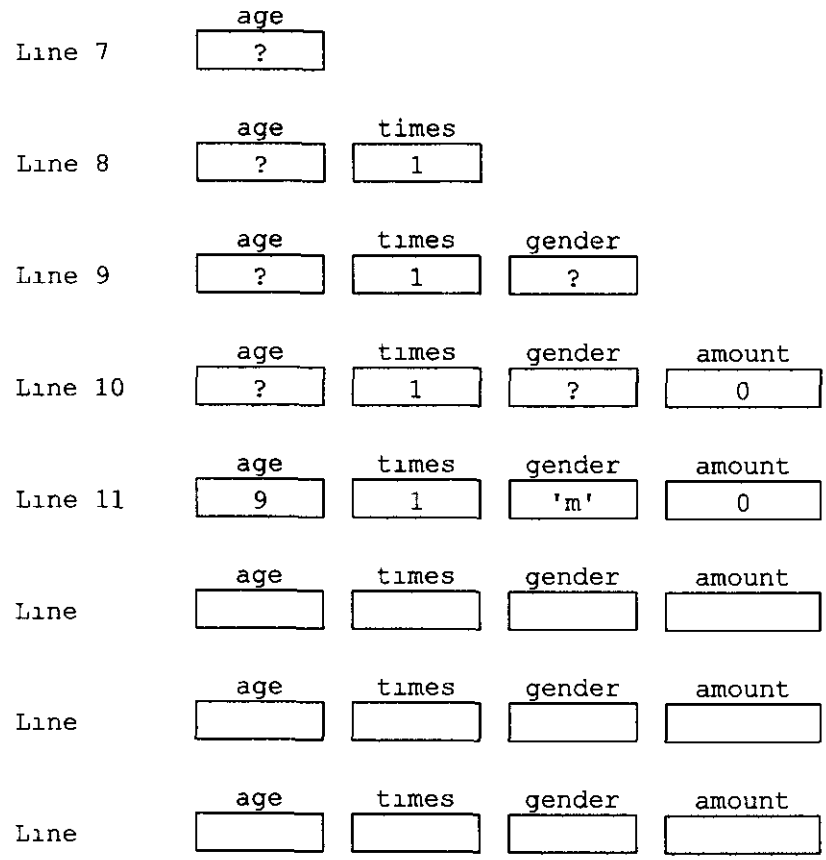
(a) [5]

Question 2(a) is based on the following program:

```
1.  #include <iostream>
2.  #include <string>
3.  using namespace std;
4.  const int GIFT = 10;
5.  int main()
6.  {
7.     int age;
8.     int times = 1;
9.     char gender; //'m' for male or 'f' for female
10.    int amount = 0;
11.    cin >> age >> gender;
12.    switch(age)
13.    {
14.      case 6:
15.      case 7:
16.        amount += 10;
17.        break;
18.      case 8:
19.        amount += 20;
20.        times = 2;
21.        break;
22.      case 9:
23.      case 10:
24.       if (gender == 'm')
25.           amount += 30;
26.        else if (gender == 'f')
27.              amount += 25;
28.      case 11:
29.        amount += 40;
30.        times = 3;
31.        break;
32.      default:
33.        cout << "Invalid age" << endl;
34.    }
35.    amount += GIFT;
36.    times += 1;
37.    cout << "amount: " << amount << " times: " << times << " gender: "
         <<  gender << endl;
```

```
38. return 0;
39. }
```

Below is part of the variable diagram for the above program We only show lines that change the value of at least one variable in the variable diagrams. Please answer the questions that follow the variable diagram:



Assume that the following input is given: `9 'm'`

(i)

Complete the three missing line numbers of the next three lines after line 11 that will change the value of at least one variable, as well as the values of the variables after these lines have been executed in your answer books. [3]

(ii)

What will the output be after line 37 has been executed? [2]

(b) [6]

Question 2(b) is based on the following program

```
 1 #include <iostream>
 2 using namespace std;
 3
 4 void productLife(string prodP, int & lifeP, int codeP)
 5 {
 6     if (prodP == "Feta cheese" || codeP == 865)
 7     {
 8        lifeP = 20;
 9        codeP = 453;
10     }
11     else if (prodP == "Yoghurt" && codeP == 658)
12     {
13         codeP = 678;
14         lifeP = 8;
15     }
16     else if (codeP == 667 || lifeP == 14)
17     {
18         lifeP = 7;
19         prodP = "Milk";
20     }
21     prodP = "Butter";
22 }
23 int main()
24 {
25     int code = 658;
26     string prod = "Feta cheese";
27     int life = 14;
28     prod = "Yoghurt";
29     productLife(prod, life, code);
30     return 0;
31 }
```

Below is part of the variable diagram for the above program. We only show lines that change the value of at least one variable in the variable diagrams. Please answer the questions that follow the variable diagram:

	[prod]	prodP	[life]|lifeP	[code]	codeP
Line 30->4	"Yoghurt"	"Yoghurt"	14	658	658
Line ?					
Line ?					
Line 21					

(i) Complete the two missing line numbers above as well as the values of the variables after these lines have been executed in your answer books, as follows:

If you think for example that the next two lines that will be executed that will change the values of one or more variables are lines 8 and 9, answer as follows in your answer book by supplying the line numbers and giving the correct values for the variables after each line has been executed respectively:

	[prod]	prodP	[life]|lifeP	[code]	codeP
Line 8	"Yoghurt"	"Feta cheese"	20	658	658
Line 9	"Yoghurt"	"Feta cheese"	20	658	453

[2]

(ii)
Complete the values of the variables after line 21 has been executed the same way as in question 2(a). [1]

(iii)
What is the value of the variable `prod` after line 29 has been executed? [1]

(iv)
What is the value of the variable `code` after line 29 has been executed? [1]

(v)
What is the value of the variable `life` after line 29 has been executed? [1]

QUESTION 3 **[8]**

The radio station *Five FM* launched a competition. There are 10 winners and the prizes are determined by the ages of the winners.

- Winners from ages 14 to 25 win a cell phone
- Winners from ages 26 to 40 win a coffee machine
- Winners over the age of 40 win a voucher at a clothing store
- There is no prize for ages below 14

The ages of the winners are input into a one-dimensional array `ages`. Complete the program below by writing down ONLY the nested `if else` statement to assign the correct value to `prize` according to the age of the winner. Do NOT copy the complete program in your answer books.

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
   string prize; //"cell phone" / "coffee machine" / "voucher" / "no prize"
   for (int k = 0; k < 10; k++)
   {
      cin >> age[k];
//
//  nested if else statement will be here
```

```
//
     cout << "For winner, aged "  << age[k] << " the prize is " <<
             prize << endl;

   }
   return 0;
}
```

QUESTION 4 **[8]**

Consider the incomplete C++ program below and answer the questions that follow. The program determines if a user is within his/or her budget or not. The program asks the user to enter the amount that he or she has budgeted for a month The program then continually prompts the user to enter each of his or her expenses for the month and keeps a running total This is repeated until a negative number is entered. Then the program should display whether the user is on, over or under budget as well as the amount with which the user is over or under budget. For example:

```
Enter your budget for the month:   400

Enter an expense, or a negative number to quit: 120
Enter an expense, or a negative number to quit: 40
Enter an expense, or a negative number to quit: 20
Enter an expense, or a negative number to quit: 130
Enter an expense, or a negative number to quit: 20
Enter an expense, or a negative number to quit: -1

You are UNDER budget by R 70
```

```
#include <iostream>
using namespace std;

int main()
{
      float budget;              // budget for the month
      float expense;             // an expense amount
      float totalExpenses;       // the total expenses
      float difference;          // amount over or under budget

      // initialize variables to 0.
      budget = 0.0;
      expense = 0.0;
      totalExpenses = 0.0;
      difference = 0.0;

      // Get the budget amount.
      cout << "Enter your budget for the month: ";
      cin >> budget;
     cout << " Enter an expense, or a negative number to quit: ";
      cin >> expense;

      // YOUR LOOP TO ENTER EACH OF EXPENSES WILL COME HERE

      // Calculate the amount over or under budget.
```

```
    difference  = budget - totalExpenses;

    // CODE TO DISPLAY WHETHER AMOUNT SPENT IS ON, OVER OR UNDER
    // BUDGET WILL COME HERE

    return 0;
}
```

(a)
Use an appropriate loop structure and write down the loop that inputs the expenses and updates the total expense amount until a value less than 0 is entered. [5]

(b)
Write down the code to determine if the amount is ON, UNDER or OVER budget and display and appropriate message with the amount. [3]

QUESTION 5 **[10]**

(a) [5]
Consider the two functions below and then answer the questions·

```
void doubleNum1 (int & valueP)
{
   valueP *=2;
}

int doubleNum2 (int valueP)
{
   valueP *= 2;
   return (valueP);
}
```

(i)
Explain in one or two sentences the difference between the two functions `doubleNum1` and `doubleNum2`.

(ii)
Is the parameter in function `doubleNum1` a value or a reference parameter? Assume the following declaration in the `main` function and give an example of a calling statement for the function `doubleNum1`.

```
    int value = 10;
```

(iii)
Is the parameter in function `doubleNum2` a value or a reference parameter? Assume the following declarations in the `main` function and give an example of a calling statement for the function `doubleNum2`.

```
    int value2 = 10;
    int result;
```

(b) [5]

Consider the incomplete program below and answer the questions that follow:

```
// cube function that returns the cube of the given integer
// YOUR cube FUNCTION CODE SHOULD COME HERE

// main program that tests the cube function
int main()
{
      int n;
      int answer;
      cin >> n;

      // CALL cube. WRITE ONLY ONE STATEMENT

      cout << answer;
      return 0;
}
```

(i) Write a function `cube` that returns the cube of the integer passed to it. For example `cube(2)` will return 8 and `cube(3)` will return 27 (because `cube(2) = 2*2*2` and `cube(3) = 3*3*3`)

(ii) The `main` function inputs a value. Use this value and write down a calling statement for the function. Use the variables already defined in `main`

QUESTION 6 [17]

A local zoo wants to keep track of how many kilograms of food each of its three monkeys eat each day during a four day period. The information must be stored in a two-dimensional array, where each row represents a different monkey and each column represents a different day of the week.

Consider the following declarations defined before the `main` function:

```
const int MONKEYS = 3;
const int DAYS = 4;
```

(a)

Declare a two-dimensional integer array called `food` with `MONKEYS` number of rows and `DAYS` number of columns. [2]

(b)

Write a function `getFoodEaten` that inputs the amount of food eaten for each monkey on each day The values are stored in the `food` array. You must use nested `for` loops to input the information Validate the input. The user must enter a value greater than 0 [7]

(c)

Write a function `displayAverageDaily` that displays the average amount of food eaten by the monkeys for each day. This is done as follows.

- For each day, calculate the total amount of food eaten

- Calculate the average eaten by dividing the total amount of food per day by the number of monkeys and display the average.
- Define the necessary variables needed for the total amount of food and the average amount of per day inside the function. Note that the average must be defined as a `float` variable

[8]

A typical session will look like this:

```
Enter the kilograms food eaten by monkey number 1
 on day 1: 3
Enter the kilograms food eaten by monkey number 1
 on day 2: 5
Enter the kilograms food eaten by monkey number 1
 on day 3: 5
Enter the kilograms food eaten by monkey number 1
 on day 4: 1
Enter the kilograms food eaten by monkey number 2
 on day 1: 1
Enter the kilograms food eaten by monkey number 2
 on day 2: 4
Enter the kilograms food eaten by monkey number 2
 on day 3: 3
Enter the kilograms food eaten by monkey number 2
 on day 4: 3
Enter the kilograms food eaten by monkey number 3
 on day 1: 5
Enter the kilograms food eaten by monkey number 3
 on day 2: 4
Enter the kilograms food eaten by monkey number 3
 on day 3: 5
Enter the kilograms food eaten by monkey number 3
 on day 4: 4
The average amount eaten on day 1 is 3.00 kilograms.
The average amount eaten on day 2 is 4.33 kilograms.
The average amount eaten on day 3 is 4.33 kilograms.
The average amount eaten on day 4 is 2.67 kilograms.
Press any key to continue . . .
```

QUESTION 7

[6]

(a)
Define a struct `ShoeType` with the following components

- `shoeStyle` (for example `'A'` of type `char`)
- `shoePrice` (for example `49.99` of type `float`)

(b)
Declare an array of type `ShoeType` called `shoes` with 5 elements.

(c)
Assume that values have already been assigned to all the elements of shoes. Complete the code segment below in order to determine the total price of all shoes of type 'B'. Write down only the necessary code in your answer book.

```
float totalPrice = 0.0;

for (int i = 0; i < 5; i++)
{
      // YOUR STATEMENTS SHOULD COME IN HERE
}

cout << "Total cost of all shoes of type B = R" << totalPrice << endl;
```

QUESTION 8 **[6]**

In this question an incomplete program is given. The main function calls a function updString that has to replace certain characters in the string and delete other characters in the string. updString has the following 4 parameters:

- string senP is the string that has to be changed
- string findP contains the string of length 3 that must be found in senP and changed
- string replaceP is the string of length 3 with which findP will be replaced
- string deleteP is the string that must be deleted from senP

For example, if the sentence

```
A big dog and a big cat ran down a street
```

is sent to the function updString with the values as in the main function below, the following will be returned to the main function:

```
A  dog andthe cat ran downthestreet
```

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4  void updString(string & senP, string findP, string replaceP,
                                                    string deleteP)
5  {
   // first replace all occurrences of findP in senP with replaceP
6      int pos = 0;
7      pos = senP.find(findP, 0);
8      while (pos != -1)
9      {
10       senP.replace(pos, 3, replaceP);
11 //Your answer to question (a) will come here
12     }
// now delete all occurrences of delete in senP
13     pos = 0;
14 //Your answer to question (b) will come here
15     while (pos != -1)
16     {
17 //Your answer to question (d) will come here
```

```
18          pos = senP.find(deleteP, pos);
19      }
20  }
21  int main(){
22      string sentence;
23  //Your answer to question (f) will come here
24      updString(sentence, " a ", "the", "big");
25      cout << sentence << endl;
26      return 0;
27  }
```

Answer the questions below to complete the program. Write down ONLY the line number and the C++ code for that line number in your answer books. Please turn the page for a number of `string` member functions to help you.

(a)
Give the correct C++ code for line 11 to find the next occurrence of `findP`.
(b)
Give the correct C++ code for line 14 to find the first occurrence of `deleteP` in `senP`.
(c)
Explain the purpose of the code in line 15 in one sentence.
(d)
Give the correct C++ code for line 17 to erase the occurrence `deleteP` found in `senP`.
(e)
Explain the purpose of the code in line 18 in one sentence.
(f)
Give the correct C++ code for line 23 to read a string into `sentence`.

A number of `string` member functions to help you

`StringObject.size( )`
 	
`StringObject.substr(startPos, length)`
 	
`StringObject.find(substring)`
`StringObject.find(substring, startPos)`
 	
`StringObject.insert(insertPos, substring);`
 	
`StringObject.erase(startPos, length);`
 	
`StringObject.replace(startPos, length, substring);`
 	
where `startPos`, `length` and `insertPos` are of type `int`, and `substring` is of type `string`

COS MEMORANDUM

NB: Please make a tick for each mark – it is required by the new assessment procedure, i.e. if the student obtains 7/10 for a question, there should be 7 ✓'s

These answers are guidelines. Please give marks for correct logic, even if answered different to the solutions. If needed, write a note to the moderator of how marks were given in this case.

Do not subtract marks for the same error twice in the same question.

SECTION A 20 marks

Question 1: 3
Question 2: 4
Question 3: 2
Question 4: 3
Question 5: 1
Question 6: 3
Question 7: 4
Question 8: 1
Question 9: 2
Question 10: 3

Each question 2 marks

SECTION B 70 marks

QUESTION 1 4 marks

(a) It reads in 3 integer values and display them in reverse order. ✓✓
If student gives a line by line explanation, e.g. "reads in a value for value3, then for value2 and then for value1 and then displays value1, value 2 and value3", only 1 mark.

(b) It counts the number of time the `string` "`red`" appears in a `string` array with 5 elements. ✓✓
If student gives a line by line explanation, only 1 mark.

QUESTION 2 11 marks

QUESTION 2a 5 marks

Note to markers: For question 2a(i) the mark is given for the highlighted field. If students missed the fact that there is no `break` statement after `case` 9, at most 3 marks are given for the question, and only if answers are correct according to the error.

(i)

	age	times	gender	amount
Line 25	9	1	'm'	30✓
Line 29	9	1	'm'	70✓
Line 30	9	3✓	'm'	70

If students missed the fact that there is no `break` statement after `case` 9, the following answer would be given:

	age	times	gender	amount
Line 25	9	1	'm'	30✓

	age	times	gender	amount
Line 35	9	1	'm'	40 ½

	age	times	gender	amount
Line 36	9	2 ½	'm'	40

If lines 35 and 36 are correct according to the error, only 1 mark must be given, otherwise none.

(ii)
`amount: 80 times: 4 gender: m` ✓✓

If students missed the fact that there is no `break` statement after `case` 9, the following answer would be given:

`amount: 40 times: 2 gender: m` ✓

Only 1 mark in this case

If students only show the variable diagrams with the values at this stage, 1 mark

QUESTION 2b 6 marks

Note to markers: For questions 2(a) and (b) the mark is given for the highlighted field. For questions 2(c) – (e), if the student left out the double quotes, but the value is correct, please give the mark.

(i)

	[prod]	prodP	[life]|lifeP	[code]	codeP
Line 13	"Yoghurt"	"Yoghurt"	14	658	678✓

	[prod]	prodP	[life]|lifeP	[code]	codeP
Line 14	"Yoghurt"	"Yoghurt"	8✓	658	678

(ii)

	[prod]	prodP	[life]|lifeP	[code]	codeP
Line 21	"Yoghurt"	"Butter"✓	8	658	678

(iii) "Yoghurt" ✓
(iv) 658 ✓
(v) 8 ✓

QUESTION 3 8 marks

Note to markers: Students must use a nested `if else` statement as specified in the question. If the student has a `cout` statement in stead of an assignment statement for `prize`, no mark.

If student did not use array subscripts, subtract 1 mark for first omission, thereafter half a mark.

If student structured answer with nested if, in such a way that '&&' can be omitted, make sure that he receives all the marks he deserves (i.e. the two marks for the &&'s should be shown somewhere.

```
            ✓           ✓           ✓
    if (age[k] >= 14 && age[k] <= 25)
        prize = "cell phone";      ½
✓                      ✓
    else if (age[k] >= 26 && age[k] <= 40)

        prize = "coffee machine";       ½

    else if (age[k] >= 41)          ✓
        prize = "voucher";          ½

    else prize = "no prize";        ½
```

QUESTION 4: 8 MARKS

Question 4a [5]

```
// Loop to enter each of expenses. Note that first expense
// was read before the loop

  ✓              ✓
while (expense >= 0) // alternative (expense > -1) /!(expense <0)
      {
          // Add the expense to the totalExpenses.
          totalExpenses = totalExpenses + expense; ✓
             //Alternatively: totalexpenses += expense

         // Get an expense amount.
         cout << "Enter an expense, or a negative number to
                        quit: ";  ✓
         cin >> expense;     ✓
      }
```

Question 4b - Give marks for any correct logic to determine if in, over or under budget [3]

```
      // Display the amount on, over or under.

      if (diff < 0) ½
      {
        // diff is negative, so make it positive for
         // display purposes and then display the status.
        diff = -(diff); //don't deduct marks if this statement
                        //not present.
        cout << "You are OVER budget by R" << diff << endl; ½
      }
      else if (diff > 0) ½
      {
         cout << "You are UNDER budget by R" << diff << endl; ½          }
      else ½
```

```
    {
        cout << "You spent the budget amount exactly.\n"; ½          }
```

QUESTION 5: 10 MARKS

Question 5a **[5]**

(i) The `void` function returns double the amount of `valueP` via the reference parameter. The `int` function returns double the amount of `valueP` through a return statement ✓
(ii) Function `doubleNum1` makes use of a reference parameter ✓ and is called as follows: `doubleNum1(value);` ✓
(iii) Function `doubleNum2` makes use of a value parameter ✓ and is called as follows: `result = doubleNum2(value2)` ✓

Question 5b **[5]**

(i)

```
int cube (int x)        ✓✓
{
     return x*x*x;          ✓
}
```

OR

```
int cube (int x)        ✓✓
{
   int value;
     value = x*x*x;         ½
     return value;          ½
}
```

OR

```
int cube (int x)        ✓✓
{
   int value;
     value = pow(x,3);              ½
     return value;          ½
}
```

(ii) **`answer = cube(n);`** ✓✓

QUESTION 6 17 marks

14]

(a) [2]

```
int food[MONKEYS][DAYS];✓✓
```

(b) [7]

```
//The getFoodEaten function gets the amount of food eaten for each //monkey on each
day. The values are stored in the food array.                          //
```

✓ ✓

```
void getFoodEaten(int food[][DAYS]) //or (int food[MONKEYS][DAYS])

{

      for (int monkey = 0; monkey < MONKEYS; monkey++) ✓
      {

            for (int day = 0; day < DAYS; day++) ✓
            {
                  // Get the amount eaten by this monkey on this day.
                  cout << "Enter the kilograms food eaten by monkey "
                        << "number " << (monkey + 1)
                        << endl << " on day " << (day + 1) << ": ";
                  cin >> food[monkey][day]; ✓

                  // Validate the input.
                  while (food[monkey][day] < 0) ✓//OR if (food[monkey][day] <0)
                  {
                        cout << "Enter a non-negative amount: ";
                        cin >> food[monkey][day]; ✓
                  }
            }
      }
}
```

(c) [8]

```
                              ✓
void displayAverageDaily(const int food[][DAYS])
                                 // or const int food[MONKEYS][DAYS])
{
      int total;   ½   // Accumulator
      float average; ½   // Average eaten
//We test here if the student realises that the array must be read   //column-wise.
Subtract one mark if array is read row by row
      for (int day = 0; day < DAYS; day++) ✓
      {
            // Initialize the accumulator.
            total = 0;  ✓

            // Get the total eaten by the monkeys on this day.
            for (int monkey = 0; monkey < MONKEYS; monkey++) ✓
                  total += food[monkey][day]; ✓

            // Calculate the average eaten.

//Only one mark if they did not realise that average is float and
//total and MONKEYS are integers
                           ½      ½                ½
            average = (float)total / (float)MONKEYS;

            // Display the average.
           cout.setf(ios::fixed);
           cout.precision(2);

// As long as the average is displayed, give the mark
           cout << "The average amount eaten on day "
                  << (day + 1) << " is " << average ½
                  << " kilograms."  <<endl;
      }
}
```

QUESTION 7 6 marks

(a)

```
struct ShoeType ✓
{
     char shoeStyle; ½
     float shoePrice; ½
};
```

(b)

```
 ShoeType shoes[5]; ✓
```

(c)

```
 for (int i = 0; i < 5; i++)
 {              ✓               ✓
     if  (shoes[i].shoeStyle == 'B')
         totalPrice += shoes[i].shoePrice; ✓

 }
```

QUESTION 8 6 marks

(a) `pos = senP.find(findP, pos + 3);` ✓
(b) `pos = senP.find(deleteP, 0);` ✓
(c) The statement makes sure that we do not read beyond the end of the string. ✓
(d) `senP.erase(pos, 3);` ✓
(e) It finds the next occurrence of `deleteP` in sentence `senP` ✓
(f) `getline(cin, sentence, '\n');` ✓

SECTION A 20 MARKS (10 MULTIPLE CHOICE QUESTIONS, 2 marks each)

Please answer this section on the **mark-reading sheet** that you received (**not** in your answer book)
Choose one option for every question

QUESTION 1

Suppose the following declarations appear in the `main` function of a C++ program.

```
float cost, fee, discount;
int nrAct;
```

If the following function header is given:

```
float totalFees(int nrActP, float feeP, float & discountP)
```

which of the options below is a correct calling statement of the function `totalFees` in the `main` function?

```
1 cost = totalFees(3, 70.00, 0.15);
2 totalFees(nrAct, fee, 0.15);
3 cost = totalFees(3, fee, discount);
4 cost = totalFees(int nrActP, float & feeP, float & discountP);
```

QUESTION 2

Suppose the following declarations appear in the `main` function of a C++ program

```
string colour, model;
int speed = 320, year = 2010;
```

If the following function header is given:

```
void showModel (int speedP, string & modelP, int & yearP, string colourP);
```

which of the options below is NOT a correct calling statement of the function `showModel` in the `main` function?

```
1 showModel (330, model, year, "blue");
2 showModel (280, "Nissan", 1999, "silver");
3 showModel (speed, model, year, colour);
4 showModel (speed, model, year, "white");
```

QUESTION 3

Suppose the following declarations appear in the `main` function of a C++ program:

```
string code;
float exam, yearmark;
string date;
bool pass;
```

Suppose the following calling statement appears in the `main` function:

```
pass = giveMark ("COS1511", 67 00, date, yearmark);
```

Which of the options below is a correct function header of the function `giveMark` in the `main` function?

```
1 bool giveMark(string & codeP, float & examP, string dateP, float yearMarkP)
2 bool giveMark("COS1511", 67.00, date, yearmark)
3 bool giveMark(codeP, examP, dateP, yearMarkP)
4 bool giveMark(string codeP, float examP, string dateP, float yearMarkP)
```

QUESTION 4

Consider the following C++ code segment and answer the question below:

```
z = 10;
cin >> x >> y;
z = y - 2 * x;
if (z <= y)
   if (z/2 > 0)
      cout << "y = " << y;
   else
      cout << "z = " << z;
   else cout << "x = " << x;
```

What will the output be for the following input?

```
2 3
```

1. There is no output
2. `z = -1`
3. `x = 2`
4. `y = 3`

QUESTION 5

Suppose the input value for `choice` is `'d'` What is the value of `x` after the following C++ code has been executed?

```
char choice;
int funny = 5;
int x = 15;
cin >> choice,
switch (choice)
{
   case 'p'  x = funny + x; break;
   case 'd': x = x / funny,
   case 'k': x = funny * funny;
   case 'm': funny = ++x - funny,
   default· funny -= x--;
}
cout << x;
```

1 25
2 26
3 3
4 15

QUESTION 6

Consider the following program

```
#include <iostream>
using namespace std;

int main()
{
   int counter = 3;
   int m = 5, n = 12;
   while (n > m)
   {
      if (n - m > 1)
         counter ++,
      n -= 2;
   }
cout << counter << " " << n << endl;
return 0;
}
```

What is displayed after this code has been executed?

1 4 6
2 3 4
3 9 5
4 3 12

QUESTION 7

Consider the following code fragment:

```
char answer;
cin >> answer;
if (answer == 'y')
   cout << "Agree" << endl;
else if (answer == 'n')
        cout << "Disagree" << endl;
     else
        cout << "No opinion" << endl;
```

Which one of the following code fragments will give exactly the same result as the code above?

1
```
char answer;
cin >> answer;
switch(answer)
{
  case 'y': cout << "Agree" << endl;
  case 'n': cout << "Disagree" << endl;
  default : cout << "No opinion" << endl;
}
```

2
```
char answer;
cin >> answer;
switch(answer)
{
  case 'y': cout << "Agree" << endl;
  case 'n': cout << "Disagree" << endl; break;
  default : cout << "No opinion" << endl;
}
```

3
```
char answer;
cin >> answer;
switch(answer)
{
  case y: cout << "Agree" << endl; break;
  case n: cout << "Disagree" << endl; break;
  default : cout << "No opinion" << endl;
}
```

4
```
char answer;
cin >> answer;
switch(answer)
{
  case 'y': cout << "Agree" << endl; break;
  case 'n': cout << "Disagree" << endl; break;
  default : cout << "No opinion" << endl;
}
```

QUESTION 8

Consider the following C++ code segment:

What will be displayed on the screen by the following code?

```
for (int count = 10, count > 0; count++)
   cout << count << " + ";
```

1 `10 + 10 + 10 + 10 + 10 +` (i e the next value will be `10` + followed by `10` + and so forth .)

2 `0 + 1 + 2 + 3 + 4 +` (i e the next value will be `5` + followed by `6` + and so forth)

3 `10 + 11 + 12 + 13 + 14 +` (i e the next value will be `15` + followed by `16` + and so forth)

4 `10 +`

QUESTION 9

Jack is throwing a costume party. You will only be allowed at Jack's party in the following cases:

- You are dressed like Cinderella and you are not wearing a hat, OR
- You are wearing your pyjamas and you are wearing formal shoes

A `bool` variable `cinderella` is `true` if you are dressed like Cinderella A `bool` variable `pjs` is `true` if you are dressed in your pyjamas A `char` variable `hat` has the value `'y'` if you are wearing a hat and `'n'` if you are not wearing a hat A `char` variable `shoes` has the value `'y'` if you are wearing formal shoes and `'n'` if you are not wearing formal shoes The Boolean variable `dressed` should receive the value `true` if your dress code satisfies the conditions given above. Which of the following options will assign the correct value to `dressed`?

1 `dressed = (cinderella && hat == 'y') || ( pjs && shoes == 'y');`

2 `dressed = (cinderella || hat == 'y') || ( pjs && shoes == 'y'),`

3 `dressed = (cinderella && hat == 'n') || ( pjs && shoes == 'y');`

4 `dressed = (cinderella && hat = 'n') || ( pjs && shoes = 'y');`

QUESTION 10

Which one of the following assignment statements is NOT a valid assignment statement in C++? You can assume that all the variables have been declared with appropriate types.

```
1  t7 =+ ++y2 - c5++,
2  x2 /= y3 - --p;
3  p8 = (x2 == 5 || y6 < 7);
4  g5 -= y3 / p / y2 / c5;
```

SECTION B 70 MARKS

QUESTION 1 [4]

In Questions 1(a) and (b) you have to write down what the purpose of the segment of code is. Look at the following example before answering the questions:

```
int a , b, c;
cin >> a >> b >> c;
cout << c << b << a;
```

The purpose of the above code segment is to input three integer values and display them in reverse order. Now answer questions 1(a) and (b) below:

(a) [2]

Explain the purpose of the following segment of code:

```
if (mark < 50)
      cout << "Failed";
else if (mark > 75)
      cout << "Distinction";
else cout << "Passed";
```

(b) [2]

Explain the purpose of the following segment of code:

```
int result[6] = {70, 61, 56, 78, 29, 87};
for (int i = 0; i < 6; i++)
   if (result[i] >= 50)
      sum += result[i];
```

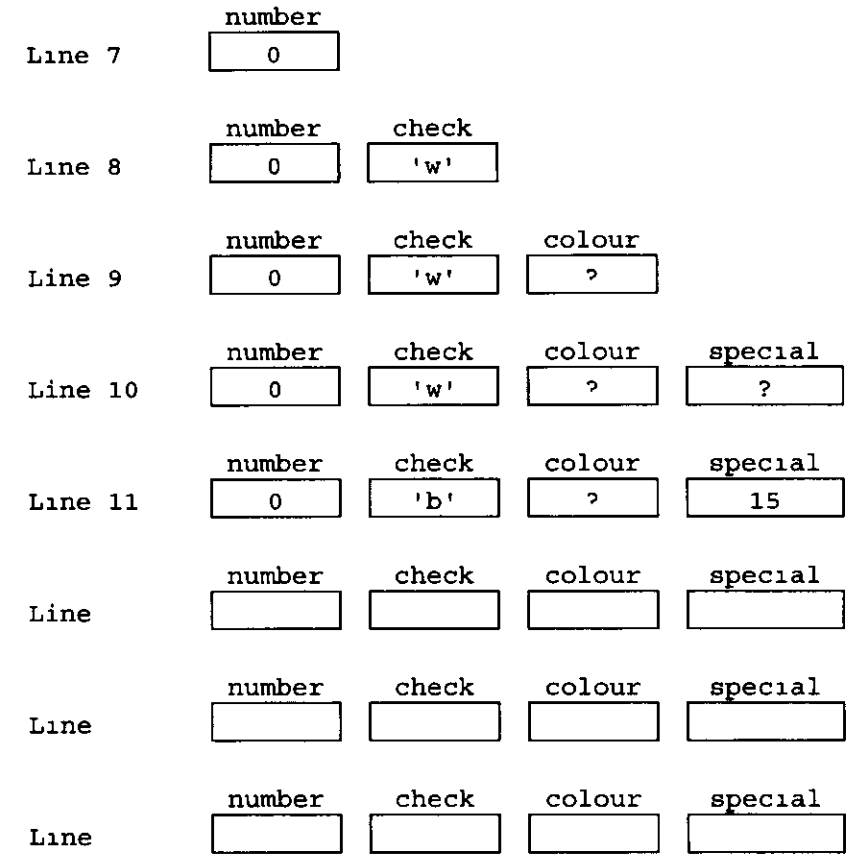
QUESTION 2 [11]

(a) [5]

Question 2(a) is based on the following program

```
1.  #include <iostream>
2.  #include <string>
3.  using namespace std;
4.  const int EXTRA = 10;
5.  int main()
6.  {
7.     int number = 0;
8.     char check = 'w';
9.     string colour;
10    int special;
11    cin >> check >> special;
12    switch (check)
13    {
14.      case 'y':
15.         colour = "yellow";
16.         number += 30;
17.         break;
18.      case 'r':
19.         colour = "red";
20.         if (special <= 5)
21.            number += special * 2;
22.         break;
23.      case 'b':
24.         colour = "blue";
25.         if (special == 20)
26.            number += special / 2;
27.         else
28.            number += special * 2;
29.      case 'g':
30.         colour = "green";
31.         break;
32.      default:
33.         cout << "invalid check field" << endl;
34.   }
35.   if (colour == "blue")
36.      number -= EXTRA;
37.   else
38.      number += 5;
39.   check = 'x';
40.   cout << "number: " << number << " colour: " << colour << endl;
41.   return 0;
42. }
```

Below is part of the variable diagram for the above program. We only show lines that change the value of at least one variable in the variable diagrams Please answer the questions that follow the variable diagram.



Assume that the following input is given b 15

(i)

Complete the three missing line numbers after line 11 for the next 3 lines that will change the value of at least one variable, as well as the values of the variables after these lines have been executed in your answer books (3)

(ii)

What will the output be after line 40 has been executed? (2)

(b) [6]

Question 2(b) is based on the following program

```
 1 #include <iostream>
 2 using namespace std;
 3
 4 void prizeBazaar(string & prizeP, int ticketP, int & ageP)
 5 {
 6     ageP = 15;
 7     if (ticketP > 200 && ticketP < 220)
 8     {
 9        prizeP = "cell phone";
10        ticketP = 0;
11     }
12     else if (ageP < 5 && prizeP == "cell phone")
```

```
13      {
14          ageP = 5;
15          prizeP = "rugby ball";
16      }
17      else if (ageP > 10 && prizeP == "pencil")
18      {
19          prizeP = "skateboard";
20          ticketP = 1;
21      }
22      ageP = 20;
23  }
24  int main()
25  {
26      int ticket = 200;
27      string prize = "pencil";
28      int age;
29
30      prizeBazaar(prize, ticket, age);
31      cout << prize << " " << ticket << " " << age;
32      return 0;
33  }
```

Below is part of the variable diagram for the above program We only show lines that change the value of at least one variable in the variable diagrams Please answer the questions that follow the variable diagram

	[prize] | prizeP	[ticket]	ticketP	[age] | ageP
Line 30->4	"pencil"	200	200	?
Line 6				
Line				
Line				

(i)
Complete the values for the variables after line 6 has been executed. (1)

(ii)
Complete the two missing line numbers above as well as the values of the variables after these lines have been executed in your answer books, as follows: (2)

If you think for example, that the two lines that will change the values of variables after line 7 has been executed, are lines 9 and 10, the following could be a possible answer. Answer as follows in your answer book by supplying the line numbers and giving the correct values for the variables after each line has been executed respectively

	[prize]|prizeP	[ticket]	ticketP	[age]|ageP
Line 9	"cell phone"	200	200	15
Line 10	"cell phone"	200	0	15

(iii)
What will the output be after line 31 has been executed? (3)

QUESTION 3 **[8]**

The incomplete C++ program below reads values into a one-dimensional array `number`. The values should then be changed as follows

- Values from 0 to 49 should be doubled
- Values from 50 to 99 should be multiplied by 10
- 100 should be subtracted from values greater than or equal to 100
- The value 0 should be assigned to values less than 0

Complete the program below by writing down ONLY the nested `if else` statement to assign the correct value to the elements of `number` according to the value that was input. Do NOT copy the complete program in your answer books.

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
   int number[20];
   for (int k = 0; k < 20; k++)
   {
      cin >> number[k],
//
// your code will be inserted here
//
      cout << "New value for number["  << k << "] is " << number[k] << endl;
}
return 0;
}
```

QUESTION 4 **[8]**

The incomplete C++ program below determines a random number and asks the user to guess what the number is If the user's guess is higher than the random number, the program displays "Too high, try again". If the user's guess is lower than the random number, the program displays "Too low, try again" If the user guesses correctly, a congratulations message is displayed Your task is to write the loop that will repeat until the user guesses correctly, or the user has had 50 tries. Use the variables that have already been defined.

For example

```
I'm thinking of a number. Can you guess what it is?
20
No, that's too low.
Guess again: 30
Sorry, that's too high.
Guess again: 25
Sorry, that's too high.
Guess again: 22
No, that's too low.
Guess again: 24
Sorry, that's too high.
Guess again  23
Congratulations! You guessed it!
```

Do not add any additional variables. Write down ONLY the missing statements.

```
include <iostream>
#include <cstdlib>    // Needed for rand and srand
using namespace std;
int main()
{
      int guess;              // The user's guess
      int count = 1;        //count the number of guesses
      int seed = 0;

      // Use the seed value to seed the random number generator.
      srand(seed);

      // Generate a random number. To keep things reasonable, we will keep it
      // within the range of 1 through 100.
      int number = 1 + rand() % 100,

      // Get the user's guess.
      cout << "I'm thinking of a number. Can you guess what it is? ";
      cin >> guess;

      // Loop to determine if the value is less than, greater than or equal
      // to the random number. YOUR CODE WILL BE INSERTED HERE

      // Congratulate the user.
```

[TURN OVER]

```
    if (count < 50)
       cout << "Congratulations! You guessed it!" << endl,
    else
    {
       cout << "Sorry, better luck next time " << endl;
       cout << "I was thinking of the number " << number << endl;
    }
    return 0;
}
```

QUESTION 5 **[10]**

(a) [5]

The function below returns a value equal to 3 times the parameter

```
int increase1(int valueP)
{
    return (valueP * 3)
}
```

(i)
Write a `void` function `increase2` that is similar to `increase1` except that it changes the value of its parameter instead of returning the result in a `return` statement

(ii)
Assume the following definitions in the `main` function

```
int increaseMe = 100;
int result;
```

Give an example of a calling statement for functions `increase1` and `increase2`

(b) [5]
Consider the incomplete program below and answer the questions that follow.

```
#include <iostream>
using namespace std;
//
// YOUR update FUNCTION CODE SHOULD COME HERE
//
// main program that tests the update function
int main()
{
   int value1, number1;
   cin >> number1 >> value1;
   update(number1, value1);
   return 0;
}
```

(i)
Write a `void` function `update` that receives two `int` parameters. The value of the second parameter must be changed in the function to the product of the first and the second parameter. For example, if the first parameter is 3 and the second parameter is 5, the function should change the second parameter to 15 because 3 * 5 = 15.

(ii)
For each parameter, state whether the parameter is a value or a reference parameter, and give a reason for your choice

QUESTION 6 **[17]**

During an entrepreneurs week, the three grade 7 teams each had a table selling all sorts of items for every day of the week from Monday to Friday. The total sales amount per day per team is saved in a two-dimensional array where each row represents a different team and each column represents a different day of the week.

Consider the following declarations defined before the `main` function:

```
const int TEAM = 3,
const int DAYS = 5;
```

(a) [2]
Declare a two-dimensional float array called sales with TEAM number of rows and DAYS number of columns

(b) [7]
Write a function `getSales` that inputs the sales amount for each team on each day. The values are stored in the sales array You must use nested for loops to input the information Validate the input to ensure that the user enters a value greater than 0 00

(c) [8]
Write a function `displayTotalDaily` that displays the total sales amount per day, as well as the total of all sales for the whole week The total sales amount per day is the sum of all the sales of all the teams for that day.

This is done as follows:

- For each day, calculate the total sales and display the total
- Keep a running total to hold the total of all sales for the whole week
- Define the necessary variables needed for the total sales amount per day and the total sales amount for the week.

A typical session will look like this.

```
Enter the sales amount for team 1 on day 1: R230
Enter the sales amount for team 1 on day 2: R160
```

```
Enter the sales amount for team 1 on day 3: R410
Enter the sales amount for team 1 on day 4: R300
Enter the sales amount for team 1 on day 5: R190
Enter the sales amount for team 2 on day 1: R340
Enter the sales amount for team 2 on day 2: R210
Enter the sales amount for team 2 on day 3: R90
Enter the sales amount for team 2 on day 4: R50
Enter the sales amount for team 2 on day 5: R111
Enter the sales amount for team 3 on day 1: R333
Enter the sales amount for team 3 on day 2: R290
Enter the sales amount for team 3 on day 3: R90
Enter the sales amount for team 3 on day 4: R70
Enter the sales amount for team 3 on day 5: R120
The total sales amount for day 1 is R903.00
The total sales amount for day 2 is R660.00
The total sales amount for day 3 is R590.00
The total sales amount for day 4 is R420.00
The total sales amount for day 5 is R421.00
The total sales for the week is R2994.00
Press any key to continue . . .
```

QUESTION 7 **[6]**

Complete the following C++ program by answering the questions below the program. Write down ONLY the line number and the code – do NOT copy the whole program in your answer books.

```
 1 #include <iostream>
 2 #include <string>
 3 using namespace std;
 4 int main()
 5 {
 6
 7
 8
 9
10
11
12     //array of struct definition follows
13
14     for (int i = 0; i < 200; i++)
15     {
16
17           cout << "Congratulations - You win a prize!" << endl;
18
19     }
20     return 0;
21 }
```

(a) **[2]**

Complete lines 6 to 11 to define a `struct` called `Balloon` that contains the following 3 variables:

- a `char` variable `size`, indicating if it is a small (`'s'`) or a medium-size (`'m'`) balloon

- a `string` variable `colour`, indicating the colour of the balloon, e g. `"red"`
- a `bool` variable `prize`, indicating if there is an extra prize to be given If `prize` is `true`, the prize will be given, else not

(b) [1]

Write down code for line 13 to define an array of structs of type `Balloon` called `bazaarBalloons` that can hold the information for 200 balloons

(c) [2]

Line 17 must only be displayed if the balloon is a medium-size balloon and if an extra prize should be given. Complete line 16 to do the necessary checking.

(d) [1]

Complete line 18 to display the colour of the current balloon

QUESTION 8 **[6]**

An incomplete C++ program is given below.The `main` function inputs a string and then calls function `updString` to change the string Function `updString` does the following:

- Checks the length of the `senP` - if it is odd, it adds the string " " to the back of `senP`
- Copies the first half of the string `senP` to the `string` variable `part1`.
- Copies the second half of the string `senP` to the `string` variable `part2`
- Finds all occurrences of string `"s"` in `part1` and replaces them with string `"z"`
- Erases all occurrences of string `" "` (that is the blank characters) from string `part2`
- Creates a new string `newString` containing `part2` followed by `part1`

For example, if the string

```
We saw six lionesses strolling across the road near the Olifants camp
```

is sent to the function, with the values as in the `main` function below, the following string will be returned

```
sstheroadneartheOlifantscamp.We zaw zix lionezzez ztrolling acro
```

The function `updString` is incomplete. Answer the questions below the program to complete the program Write down ONLY the line number and the C++ code for that line number in your answer books.

```
 1 #include <iostream>
 2 #include <string>
 3 using namespace std;
 4 string updString(string & senP)
 5 {
 6     int len = 0;
 7     int pos = 0;
 8     string newstring, part1, part2;
 9     len = senP.size();
10     if (len % 2 == 1)
```

```
11      {
12         senP += ".",
13         len +=1,
14      }
15 //Your answer to question (a) will come here
16      part2 = senP.substr(len/2, len/2);
17 //replace all "s" with "z" in the first half of the string
18      pos = part1.find("s", 0);
19      while (pos != -1)
20      {
21 //Your answer to question (c) will come here
22        pos = part1.find("s", pos + 1),
23      }
   //delete all " " (space characters) in the second half of the string
24      pos = 0;
25 //Your answer to question (d) will come here
26      while (pos != -1)
27      {
28        part2.erase(pos, 1),
29        pos = part2.find(" ", pos);
30      }
   //copy first the changed second half and then the changed first half
   //to newstring
31 //Your answer to question (e) will come here
32        return newstring;
33 }
34 int main(){
35      string sentence, newSentence;
36      getline(cin, sentence, '\n');
37 //Your answer to question (f) will come here
38      cout << newSentence << endl;
39      return 0;
40 }
```

(a) Give the correct C++ code for line 15 to assign the first half of `senP` to `part1`.

(b) Explain the purpose of the code in line 19 in one sentence

(c) Give the correct C++ code for line 21 to replace the occurrence of `"s"` found in `part1`, with `"z"`

(d) Give the correct C++ code for line 25 to find the first occurrence of `" "` (the blank character) in `part2`

(e) Give the correct code for line 31 to assign `part2` followed by `part1` to `newString`

(f) Give the correct code for line 37 to call function `updString` with variable `sentence` as parameter. Variable `newSentence` should receive the changed string

A number of `string` member functions to help you

```
StringObject.size( )

StringObject.substr(startPos, length)

StringObject.find(substring)
StringObject.find(substring, startPos)

StringObject.insert(insertPos, substring);

StringObject.erase(startPos, length);

StringObject.replace(startPos, length, substring);
```

where

`startPos`, `length` and `insertPos` are of type `int`, and
`substring` is of type `string`

NB: Please make a tick for each mark – it is required by the new assessment procedure.

SECTION A 20 marks

Question 1: 3
Question 2: 2
Question 3: 4
Question 4: 2
Question 5: 1
Question 6: 1
Question 7: 4
Question 8: 3
Question 9: 3
Question 10: 1

Every question 2 marks

SECTION B 70 marks

QUESTION 1 4 marks

(a) It displays a message according to the mark 2
If student gives a line by line explanation, only 1 mark.

(b) Adds all the elements in an array with a value >= 50 2
If student gives a line by line explanation, only 1 mark.

QUESTION 2 11 marks

QUESTION 2a 5 marks

Note to markers: For question 2a(i) the mark is given for the highlighted fields. If students missed the fact that there is no `break` statement after `case` 9, at most 1 mark is given for (ii) if correct according tot the error.

(i)

	number	check	colour	special
Line 24	0	'b'	"blue"	15
Line 28	30	'b'	"blue"	15
Line 30	30	'b'	"green"	15

[3]

If students missed the fact that there is no `break` statement after `case` 'b', the following answer would be given:

	number	check	colour	special
Line 24	0	'b'	"blue"	15
Line 28	30	'b'	"blue"	15
Line 30	20	'b'	"blue"	15

No mark for line 36, but 1 mark can be given in (ii) below if correct according to this error

(ii)

`number: 35 colour: green` ✓✓

If students missed the fact that there is no `break` statement after `case` 9, the following answer would be given:

`number: 20 colour: blue` ✓

Only 1 mark in this case

Question2(b)

Note to markers: For questions 2(a) and (b) the mark is given for the highlighted field. For questions 2(c) , if the student left out the double quotes, but the value is correct, please give the mark.

(a)

	[prize]|prizeP	[ticket]	ticketP	[age]|ageP
Line 6	"pencil"	200	200	15

[1]

(b)

	[prize]|prizeP	[ticket]	ticketP	[age]|ageP
Line 19	"skateboard"	200	200	15
Line 20	"skateboard"	200	1	15

[2]

(c) `skateboard 200 20` [3]

QUESTION 3 8 marks

Note to markers: Students must use a nested `if`…`else` statement. No marks if `cout` was used instead of assignment statements

```
                ✓       ✓           ✓
  if (number[k] >= 0 && number[k] <= 49)
     number[k] *= 2;          ½
else if (number[k] >= 50 && number[k] <= 99) ✓✓
     number[k] *= 10;         ½
else if (number[k] >= 100)    ✓
     number[k] -= 100;        ½
else number[k] = 0;           ½
```

Students could also have put the cout statement within each case statement. One mark overall for the cout.

QUESTION 4 8 marks

```
 ✓              ✓                ✓
while (guess != number && count < 50)
     {


          if (guess < number) ✓
          {
            cout << "No, that's too low." << endl; ½
          }
          else if (guess > number) ✓
          {
            cout << "Sorry, that's too high." << endl; ½
          }

          // Get another guess.
          cout << "Guess again: ";
          cin >> guess;             ✓
          count += 1;               ✓
          }
```

QUESTION 5: 10 MARKS

(a) [5]

(i)

```
void increase2(int & value) ✓
{
   value *= 3; ✓
}
```

(ii)

```
result = increase1(increaseMe); ✓✓
increase2(increaseMe);  ✓
```

(b) [5]

(i)

```
                  ✓           ✓
void update(int number, int & value)
{
    value *= number; ✓
}
```

(ii)

`number` should be a value parameter because the function should not change it. ✓

`value` should be a reference parameter, because the function changes its value. ✓

QUESTION 6 18 marks

(a) [2]

```
float sales[TEAM][DAYS] ✓✓
```

(b) [7]

```
// The sales function reads in the total sales amount for each team
for each day of the week
//
 ✓                          ✓
void getSales(float sales[][DAYS]) //or int sales[TEAM][DAYS]
{
      for (int row = 0; row < TEAM; row++)      ✓
      {
         for (int col = 0; col < DAYS; col++)   ✓
         {
            // Get the sales amount for this team
            cout << "Enter the sales amount for team "
                 << (row + 1) << " on day " << (col + 1) << ": R";
                  cin >> sales[row][col]; ✓

                  // Validate the input.
                  while (sales[row][col] < 0)   ✓
                  {
                        cout << "Enter a non-negative amount: ";
                        cin >> sales[row][col]; ✓
                  }
         }
      }
}
```

(c) [9]

```
                      ✓                ✓
void displayTotalDaily(const float sales[TEAM][DAYS])
{
      float total;  // total sales per day      ½
      float weekTotal = 0.00;  // total sales for the whole week   ½

   //Only 1 mark for the nested if, if read row by row in stead of
   //column by column
      for (int col = 0; col < DAYS; col++)      ✓
      {
            // Initialize the total.
            total = 0.0;            ✓
```

```
            // Get the total sales for this day.
            for (int row = 0; row < TEAM; row++) ✓
                  total += sales[row][col];      ✓
                  weekTotal += total;            ✓

          // Display the total.
           cout.setf(ios::fixed);
           cout.precision(2);

         cout << "The total sales amount for day "
                 << (col + 1) << " is R" << total << endl; ½


    } //
    //display total sales for the week                        ½
         cout << "The total sales for the week is R" << weekTotal
              << endl;
}
```

QUESTION 7 6 marks

(a)

```
 6      struct Balloon     ½
 7      {
 8          char size;      ½
 9          string colour;  ½
10          bool prize;     ½
11      };
```

(b)

```
Balloon bazaarBalloon[200]; ✓
```

(c)

```
                          ✓                          ✓
if (bazaarBalloon[i].prize && bazaarBalloon[i].size == 'm')
                         OR
                          ✓                          ✓
if (bazaarBalloon[i].prize == true && bazaarBalloon[i].size == 'm')
```

(d)

```
                         ✓
cout << bazaarBalloon[i].colour << endl;
```

QUESTION 8 6 marks

(a) `part1 = senP.substr(0 , len/2);` ✓

(b) It makes sure that we do not read past the end of the sentence ✓

(c) `part1.replace(pos, 1, "z");` ✓

(d) `pos = part2.find(" ", 0);` ✓

(e) `newstring = part2 + part1;` ✓

(f) `newSentence = updString(sentence);` ✓