Grading Guidelines
End-Semester Test

QA1 – QA4 [Corrected by AD]

Part (a): 0 for incorrect formula. Penalty of 1 or 2 if there are syntax errors (like missing *, or using [] instead of ()).

Part (b):

Marks breakup

QA1:	[A] 1	[B] 2	[C] 2	[D] 3
QA2:	[A] 2	[B] 2	[C] 2	[D] 2
QA3:	[A] 2	[B] 2	[C] 2	[D] 2
QA4:	[A] 2	[B] 2	[C] 3	[D] 1

QA1:

[C] 1 mark for the condition, 1 mark for the recursive call (and update of count)
[D] 1 mark for each of the two conditions, 1 mark for the recursive call (and update of count)

Common errors: Setting count = … (instead of count += …), return the result from recursive call (you need to make multiple recursive calls, so return will not work), use of lastadded for all colors (only red balls cannot be repeated), missing/wrong color as the last parameter.

QA2:

[A] 1 mark for each conditions
[C] 1 mark for the condition, 1 mark for the recursive call (and update of count)
[D] 1 mark for the condition, 1 mark for the recursive call (and update of count)

Common errors: Setting count = … (instead of count += …), return the result from recursive call (you need to make multiple recursive calls, so return will not work), using bad or missing greenlast values in recursive calls.

QA3:

[B] 1 mark for the recursive call, 1 mark for the update of count (count = … works in [B], not in [C] or [D])
[C] 1 mark for the recursive call, 1 mark for updating count
[D] 1 mark for the recursive call, 1 mark for updating count

Common errors: Setting count = … (instead of count += …), return the result from recursive call (you need to make multiple recursive calls, so return will not work), using b-1 in the recursive call of [D] (you cannot add blue balls one by one, this will violate all blue balls coming together).

QA4:

[A] 1 mark for the recursive call, 1 mark for the update of count (count = … works in [B], not in [C] or [D])
[B] 1 mark for the condition, 1 mark for the recursive call (and update of count)
[C] 1 mark for each of the two conditions, 1 mark for the recursive call (and update of count)

Common errors: Setting count = … (instead of count += …), return the result from recursive call (you need to make multiple recursive calls, so return will not work), a check r == 0 is redundant in [C] (not penalized), 0 or the value from a recursive call in [D].

General remark: 0 awarded in Part (b) if you have answered a question not given to you.

QF1 – QF4 [Corrected by AD]

Part (a): 0 for incorrect formula (1 if only the sign of the constant is incorrect). Penalty of 1 or 2 if there are syntax errors (like using [] instead of ()).

Part (b):

Marks breakup

QF1:	[A] 1	[B] 1	[C] 3	[D] 3
QF2:	[A] 1	[B] 3	[C] 3	[D] 1
QF3:	[A] 1	[B] 1	[C] 3	[D] 3
QF4:	[A] 1	[B] 3	[C] 3	[D] 1

QF1:

[C] 1 mark for the condition, 1 mark for the recursive call, 1 mark for the update of count
[D] 1 mark for the condition, 1 mark for the recursive call, 1 mark for the update of count

Common errors: Setting count = … (instead of count += …), return the result from recursive call (you need to make multiple recursive calls, so return will not work), decrement of N in recursive calls (this is bad because in the end you check n == N).

QF2:

[B] 1 mark for the condition, 1 mark for the recursive call, 1 mark for the update of count
[C] 1 mark for the condition, 1 mark for the recursive call, 1 mark for the update of count

Common errors: Setting count = … (instead of count += …), return the result from recursive call (you need to make multiple recursive calls, so return will not work), non-palindromic addition of summands (do not add 1 or 2 or 3 individually, and make a recursive call on n-1, n-2, or n-3. This will not guarantee palindromic sums.), return 2 is valid in [D] but not any other constant.

QF3:

[C] 1 mark for the condition, 1 mark for the recursive call, 1 mark for the update of count
[D] 1 mark for the condition, 1 mark for the recursive call, 1 mark for the update of count

Common errors: Setting count = … (instead of count += …), return the result from recursive call (you need to make multiple recursive calls, so return will not work), passing 1 or 0 in all of added1, added2, and added3. These three are flags taking only 0 or 1 as their values, some used these variables as counts. This is not penalized if consistently done throughout the code.

QF4:

[B] 1 mark for each of the two conditions, 1 mark for the recursive call (and the update of count)
[C] 1 mark for the condition, 1 mark for the recursive call, 1 mark for the update of count

Common errors: Setting count = … (instead of count += …), return the result from recursive call (you need to make multiple recursive calls, so return will not work), the check lastsmd <= 3 is redundant in [C] (not penalized, not any extra credit for writing it), passing lastsmd (or expressions like lastsmd + 1) as the second argument in [B] or [C] (in [A] lastsmd is fine).

General remark: 0 awarded in Part (b) if you have answered a question not given to you.

QB1 – QB3 [Corrected by AG]

- Guidelines are shown below only for commonly found answers/mistakes. Marks are deducted/given for many other things specific to an answer.

- **Part (a):** 2 if correct, 0 if not. There is no partial marking.

- **Part (b):** Each blank has 2 marks.

    - **QB1**
        - For [A], 0.5 mark for return value type, 1 mark for A, 0.5 marks for *idx (any other name for idx , or the same parameters in any other order is fine also)
        - For [B], 1 mark for strcmp() > 0 syntax, 1 mark for correct parameters of strcmp
        - For [C], 1.5 mark for malloc, 0.5 marks for new = .).5 deducted if you missed the +1 inside
        - For [D], 1 mark for strcpy, 1 mark for correct return of *idx
        - Some of you have mixed [C] and [D]. Marks given as long as the malloc is done before the strcpy

    - **QB2**
        - For [A], 0.5 mark for return value type, 1 mark for X, 0.5 marks for *idx (any other name for idx , or the same parameters in any other order is fine also)
        - For [B], 1 mark for strcmp() > 0 syntax, 1 mark for correct parameters of strcmp
        - For [C], 1.5 mark for malloc, 0.5 marks for new = .).5 deducted if you missed the +1 inside
        - For [D], 1 mark for strcpy, 1 mark for correct return of *idx
        - Some of you have mixed [C] and [D]. Marks given as long as the malloc is done before the strcpy

    - **QB3**
        - For [A], 0.5 mark for return value type, 0.5 mark for P, 0.5 marks for *cnt, 0.5 mark for S (any other name for cnt , or the same parameters in any other order is fine also)
        - For [B], 1 mark for strcmp() > 0 syntax, 1 mark for correct parameters of strcmp
        - For [C], 1.5 mark for malloc, 0.5 marks for new = .).5 deducted if you missed the +1 inside
        - For [D], 1 mark for strcpy, 1 mark for correct return of *cnt

- Some of you have mixed [C] and [D]. Marks given as long as the malloc is done before the strcpy

QG1 – QG3 [Corrected by AG]

- Guidelines are shown below only for commonly found answers/mistakes. Marks are deducted/given for many other things specific to an answer.

- **Part (a):** For QG1 and QG2, 1 mark given only if first or last part of the array upto middle is ok, other part is wrong. For QG3, 2 if correct, 0 if not. There is no partial marking for QG3.

- **Part (b):** Each blank has 2 marks.

    - **QG1**
        - For [A], 1 mark given for declaration of A, 1 mark given for **newarr (or any other name). Same parameters in any other order is also ok. 0.5 deducted if *newarr is done instead of **newarr
        - For [B], 1.5 marks for malloc, 0.5 for new =
        - For [C], 1 mark given for sum +=, 1 mark for right hand side
        - For [D], 0.5 deducted if * is missed but otherwise correct

    - **QG2**
        - For [A], 1 mark given for declaration of X, 1 mark given for **newarr (or any other name). Same parameters in any other order is also ok. 0.5 deducted if *newarr is done instead of **newarr
        - For [B], 1.5 marks for malloc, 0.5 for new =
        - For [C], 1.5 marks for rhs of >, 0.5 for lhs
        - For [D], 0.5 deducted if * is missed but otherwise correct

    - **QG3**
        - For [A], 1 mark given for declaration of P, 1 mark given for **arr (or any other name). Same parameters in any other order is also ok. 0.5 deducted if *arr is done instead of **arr
        - For [B], 1 mark for sum+=, 1 mark for rest
        - For [C], 1.5 marks for malloc, 0.5 for lhs of =
        - For [D], 1 mark for copyArr, 1 mark for *arr = p. 0.5 deducted if arr is used instead of *arr
        - Some of you have mixed [C] and [D]. Marks given as long as the malloc is done before the copyArr
        - Some of you passed a 1-d array and copied the row there. Marks deducted as you had to return the new array, not copy it into another array passed. If you had to copy, there would be no need of the new array, or malloc etc., could have copied directly from the row of the 2-d array.

QC1 – QC4 [Corrected by DRC]

- Guidelines are shown below only for commonly found answers/mistakes. Marks are deducted/given for many other things specific to an answer.
- If you have cheated in any of the parts (a) or (b), you have been given a 0 for the entire question.
  - **Part (a):** There are two terms. Each term is of 1 mark. There is no partial marking.

    Common mistakes:

    - If the correct answer is B.ptr->ptr->Rank and either (i) "." Is replaced by -> or (ii) -> is replaced by ".", then you have been awarded 0.5 marks, provided remaining portions are correct.
    - If you were asked to use only struct B and have used any other struct in any term, no marks have been awarded for that term.
  - **Part (b):** Each blank has 2 marks.

- There is no partial marking for the blanks [A], [B], [C].
- For blank [A], you have been awarded 0 marks for putting "<" instead of ">" or vice versa since that produces wrong result.
- For blank [C], you have been awarded 0 marks for putting wrong row and column indices; e.g if A[i][j] is written instead of the correct answer A[j][i].
- There are three items to be printed in blank [D]. You have been awarded (i) 1.5 marks if any two are correct and (ii) 0.5 if only one is correct. You have been awarded 0 if item(s) are correct but ordering of items are wrong since that gives wrong output.
- If there is a mistake/typo in the variable names, 0.5 marks has been deducted for each such mistake. (Please note that if you have made a typo, it is a mistake and it will be treated as such). However, if the variable names used are same as the variable names of any other alternate question, then you have been awarded 0 marks.

QH1 – QH4 [Corrected by DRC]

· Guidelines are shown below only for commonly found answers/mistakes. Marks are deducted/given for many other things specific to an answer.

· If you have cheated in any of the parts (a) or (b), you have been given a 0 for the entire question.

· **Part (a):** There are four terms. Each term is of 0.5 marks. There is no partial marking.

Common mistakes:

- If we have "struct polyterm A, B" and A has a member "coeff", "A->coeff" is not defined. It should be "A.coeff".
- If A has a member pointer "ptr" that points to struct B, then "*(A.ptr).coeff" produces an error. The correct use has brackets - "(*(A.ptr)).coeff".
- If you were asked to use only struct A and have used any other struct in any term, no marks have been awarded for that term.

· **Part (b):** Each blank has 2 marks.

Common mistakes:

- If you were asked to print along rows and you have printed along columns (or vice versa), you have been awarded 0.
- Blank [B] has the most mistakes.
  - The if condition cannot be checked using the row/column index alone, you will need to calculate the difference of the total number of rows (columns) and the row (column) index.
  - If you have checked this difference but have made some other error, 1 mark has been awarded.
  - Otherwise, 0 has been given for blank [B].
- If all blanks other than blank [B] are correct, you get 6 marks for part (b).
- If blanks [A] and [B] are correct, but blanks [C] and [D] have been interchanged, you get 6 marks for part (b).
- Blanks [A], [C] and [D] each had the syntax of a for loop as answer. For each of them -
  - If there is a mistake in the checking condition and everything else is correct, 1 mark is given for that blank.
  - If there is a mistake/typo in the initialization or incrementing, 0.5 marks has been deducted for each such mistake. (Please note that if you have made a typo, it is a mistake and it will be treated as such).
- Mixing up the "i" and "j" indices. Since the question prints M[i][j], interchanging the indices does not work.
  - If everything else is correct, 4 marks are given in total for part (b).
  - If blank [B] is also wrong, 2 marks are given in total for part (b).

QD1 – QD3 [Corrected by DSM]

QD1:

Part (a): Correct answer:

[A] &(t -> age)

[B] t -> age

Each part carry mark 1. Binary grading: 1 for each correct answer and 0 for wrong answer.

Part (b): The question is to find **intersection of two linked lists.**

[A] : Correct answer is

tmp1->x == tmp2->x

Full marks is 2 for correct answer. Binary grading. No other variable is allowed.

[B] : Correct answer is

tmp2 = tmp2->link

Full marks is 2 for correct answer. Use of another variable (not defined in the program are not accepted. Binary grading.

[C] : Correct answer is

tmp2 != NULL

Full marks is 2 for correct answer. Binary grading. No other variable/ statement is allowed.

[D] : Correct answer is

new->link = L; L = new;

Full mark is 2. Partial marking if the blank statements are given partially. Insert at end or insert at front case is considered as correct.

QD2:

Part (a): Correct answer:

[A] s.name

[B] s.name

Each part carry mark 1. Binary grading: 1 for each correct answer and 0 for wrong answer.

Part (b): The question is to find **union of two linked lists.**

[A] : Correct answer is

p1 -> x == p2 -> x

Full marks is 2 for correct answer. Binary grading. No other variable is allowed. No other logic with multiple statement is allowed.

[B] : Correct answer is

p2 = p2 -> next

Full marks is 2 for correct answer. Use of another variable (not defined in the program are not accepted. Binary grading. No other logic with multiple statement is allowed.

[C] : Correct answer is

p2 == NULL

Full marks is 2 for correct answer. Binary grading. No other variable/ statement is allowed. No other logic with multiple statement is allowed.

[D] : Correct answer is

q -> next = L; L = q;

Full mark is 2. Partial marking if the blank statements are given partially. Insert at end or insert at front case is considered as correct. Use of other variable/ statement is not allowed. No other logic with multiple statement is allowed.

QD3:

Part (a): Correct answer:

{ "zyzzyva", "insect" }

Partial grading if syntax of initialization is with minor difference than the actual. Initialization with other string is not allowed.

Part (b): The question is to find **difference between L1 and L2 (i.e. element in L1 but not in L2).**

[A] : Correct answer is

(p != NULL) && (p -> x != L1 -> x)

Full marks is 2 for correct answer. Partial grading if partial condition checking. No other variable is allowed. No other logic with multiple statement is allowed.

[B] : Correct answer is

p == NULL

Full marks is 2 for correct answer. Use of another variable (not defined in the program are not accepted. Binary grading. No other logic with multiple statement is allowed.

[C] : Correct answer is

n -> ptr = L; L = n;

Full marks is 2 for correct answer. Partial grading if partial statement. No other variable is allowed. No other logic with multiple statement is allowed.

[D] : Correct answer is

L1 = L1 -> ptr;

Full marks is 2 for correct answer. Use of another variable (not defined in the program are not accepted. Binary grading. No other logic with multiple statement is allowed.

QI1 – QI3 [Corrected by DSM]

QI1:

Part (a): Correct answer: "%c%c", s.roll[0], s.roll[1]
Binary. 2 marks for correct answer and 0 for wrong answer.

Part (b): The question is to create a new list T containing **every third element** from L **starting from the third element**.

[A] : Correct answer is
(L==NULL)||(L->next==NULL)||(L-> next -> next == NULL)
Full marks is 2 for correct answer. Part credit is awarded for correct but partial condition checking.

[B] : Correct answer is
L -> next -> next
Full marks is 2 for correct answer. Use of another variable (not defined in the program are not accepted. The pointer movement in two different steps are also considered as correct.
No credit if only one shifting is given.

[C] : Correct answer is
if(L!=NULL) L=L->next; if(L!=NULL) L=L->next;
Full mark is 2 for correct answer.
Partial marking for (part) movement or no null point checking.

[D] : Correct answer is
p -> next = new; p = p -> next;
Full mark is 2.
Insert at end or insert at front cases are considered as correct.

QI2:

Part (a): Correct answer: "%c%c", myfriend.roll[2], myfriend.roll[3]
Binary grading. 2 marks for correct answer and 0 for wrong answer.

Part (b): The question is The question is to create a new list T containing **every third element** from L **starting from the second element**.

[A] : Correct answer is
(L == NULL) || (L -> link == NULL)
Full marks is 2 for correct answer. Part credit is awarded for correct but partial condition checking. No mark, if wrong variables are referred.

[B] : Correct answer is
if (L!=NULL) L=L->link; if (L != NULL) L = L -> link;
Full marks is 2 for correct answer. Use of another variable (not defined in the program are not accepted.
Partial marking for (part) movement or no null point checking.

[C] : Correct answer is
L == NULL
Full mark is 2 for correct answer. Binary marking.

[D] : Correct answer is
p -> next = new; p = p -> next;
Full mark is 2.
Insert at end or insert at front cases are considered as correct.

QI3:

Part (a): Correct answer: thatstud.roll[4]
Binary grading. 2 marks for correct answer and 0 for wrong answer.

Part (b): The question is The question is to create a new list T containing **every third element** from L **starting from the fourth element**.

[A] : Correct answer is
L == NULL

Full marks is 2 for correct answer. No mark, if wrong variables are referred or pointer movement is wrong.

[B] : Correct answer is

T = q = new;

Full marks is 2 for correct answer. Use of another variable (not defined in the program are not accepted.

Partial marking for one correct statement.

[C] : Correct answer is

q -> ptr = new; q = q -> ptr;

Full mark is 2 for correct answer. Use of another variable (not defined in the program are not accepted.

Partial marking for one correct statement.

[D] : Correct answer is

if (L!=NULL) L=L->ptr; if(L!=NULL) L=L->ptr;

Full mark is 2.

Full marks is 2 for correct answer. Use of another variable (not defined in the program are not accepted.

Partial marking for (part) movement or no null point checking.

QE1 – QE3 and QJ1 – QJ3 [Corrected by SG]

Gradings for all blanks are binary except QE1, Part (b), [C] and [D] which have two parts with 1 mark each.

QJ1, Part (b) also has an alternate solution as follows which is accepted.

```
[B] if (A[i] > B[i]) gf ++;
[C] if (A[i] < B[i]) lf ++;
[D] ((gf == 0) && (lf > 0)) || ((gf > 0) && (lf == 0))
```