1. Let $A$ be an array containing $n$ integer values. Consider the following algorithm for deciding whether there is no value that appears in at least one half of the entries of array $A$. For simplicity, assume that $n$ is even.

Algorithm Check($A, n$)
In: array $A$ storing $n$ integer values.
Out: true if no value in $A$ appears in at least $n/2$ entries of $A$; false otherwise.

$x = \text{random value from } A$
$c = 0$
for $i = 0$ to $n - 1$
    if $A[i] = x$ then $c = c + 1$
if $c < n/2$ then return true
else return false

• (3 marks) Show that this algorithm sometimes produces an incorrect output.
• (5 marks) Compute the maximum probability that the algorithm produces a wrong output.
• (5 marks) Modify the algorithm so that the probability that it produces a wrong output is at most $(\frac{1}{2})^{100}$.
• (7 marks) Prove that your algorithm gives the wrong answer with at most the above probability.

2. Consider a cloud computing system with $n$ computers. Throughout the day the administrator of the system receives requests from clients asking for the use of a certain number of computers. Each request $(c, p)$ consists of the number $c$ of computers required and the price $p$ that the user is willing to pay per computer. When a request is received the administrator must decide right away (without knowing which requests will arrive in the future) whether it should be accepted or rejected. If the request is accepted, then the required number of computers is assigned to the client for the rest of the day. At the end of the day all the allocations are released, so that at the beginning of the following day the $n$ computers are available to process new requests. The goal is to maximize the daily profit.

Assume that for each request $(c, p)$, the price that a client is willing to pay per computer is at least 1 and at most 6. Moreover, assume that a client is never allowed to request more than $n/k$ computers per request for some fixed value $k$ selected by the administrator, or in other words, for each request $(c, p)$, $c \leq n/k$.

• (5 marks) Write an online algorithm with constant competitive ratio for the above problem. Assume that $k \geq 2$ is integer and that $n$ is a multiple of $k$.

Use the following notation to write your algorithm. Let $C_A$ be the number of computers available and $P_E$ be the total profit earned. At the beginning of the day $C_A = n$ and $P_E = 0$. The values of $C_A$ and $P_E$ change as the algorithm processes each request. Complete the following algorithm:

Algorithm Process $(c, p)$
In: request $c, p$
Out: true if the request is accepted; false otherwise

• (10 marks) Compute the competitive ratio of your algorithm and show that it is constant. Explain your answer.

3. You are looking to hire a company to complete some project for you. You contact several companies and each one of them sends a representative to explain you how much they will charge you to complete the project. Representatives come at different times, and each one of the them makes you an offer which
is cancelled as soon as the representative leaves you office; hence, after talking with a representative you need to decide right away whether you will hire their company or not. You know that there will be $n$ companies visiting you. You also know that the minimum amount of money that they can charge you to complete the project is $0.5M and the maximum amount is $8M.

- (7 marks) Write an online algorithm to determine which offer you will accept that has competitive ratio at most 4. To answer this question, complete the following algorithm:

  Algorithm \textsc{Project} \hspace{1em} (offer, offer number)
  \hspace{1em}\textbf{In:} offer: value a company is offering to charge, offer number: number of offers that have been received, including the current one.
  \hspace{1em}\textbf{Out:} true if this offer is accepted; false otherwise.

- (8 marks) Show that your algorithm has competitive ratio at most 4.