Consider a programming language that consists of commands $C$ composed from assignments $X := E$ (where $X$ is a program variable, and $E$ is an arithmetic expression), heap allocation $X := \text{alloc}(E_1, \ldots, E_n)$, heap assignment $[E_i] := E_2$, heap dereference $X := [E]$, disposal of heap locations $\text{dispose}(E)$, the no-op $\text{skip}$, sequencing $C_1; C_2$, conditionals $\text{if } B \text{ then } C_1 \text{ else } C_2$ (where $B$ is a boolean expression), and loops $\text{while } B \text{ do } C$. $\text{null}$ is 0

(a) Explain informally what it means for a separation logic partial correctness triple \{P\} $C$ \{Q\} to be valid. [3 marks]

(b) Explain informally what it means in terms of the executions of $C$ for the separation logic partial correctness triple \{⊤\} $C$ \{⊥\} to be valid. [2 marks]

(c) Recall the list representation predicate $\text{list}$:
\[
\text{list}(t, []) = (t = \text{null}) \quad \text{list}(t, h :: α) = \exists y. ((t \mapsto h) \ast ((t + 1) \mapsto y) \ast \text{list}(y, α))
\]

We write [] for the empty mathematical list; $h :: α$ for the mathematical list the head of which is $h$, and the tail of which is $α$; $α ++ β$ for the concatenation of mathematical lists $α$ and $β$; $α[i]$ for the $i$-th element of the list $α$, starting at 0; and $[k, \ldots, n]$ for the ascending list of integers from $k$ to $n$, including $k$ and $n$.

Give a proof outline, including a loop invariant, for the following triple:
\[
\{N = n \land N \geq 0\} \quad X := \text{null}; \text{while } N > 0 \text{ do } (X := \text{alloc}(N, X); N := N - 1) \quad \text{list}(X, [1, \ldots, n]!)
\]

[4 marks]

(d) Also recall the partial list representation predicate $\text{plist}$:
\[
\text{plist}(t, [], u) = (t = u) \\
\text{plist}(t, h :: α, u) = \exists y. ((t \mapsto h) \ast ((t + 1) \mapsto y) \ast \text{plist}(y, α, u))
\]

Give a loop invariant for the following list sum triple:
\[
\text{list}(X, α) \land N = \sum_{i=0}^{\text{length}(α) - 1} α[i]
\]

[4 marks]

(e) Give a loop invariant for the following list concatenation triple:
\[
\text{list}(X, α) * \text{list}(Y, β)
\]
\[
\text{if } X = \text{null} \text{ then } Z := Y \text{ else }
\quad (Z := X; U := Z; V := [Z + 1];
\quad \text{while } V \neq \text{null} \text{ do } (U := V; V := [V + 1]);
\quad [U + 1] := Y)
\quad \text{list}(Z, α ++ β)
\]

[5 marks]

(f) Describe precisely a stack and a heap that satisfy $\text{list}(X, [1, \ldots, 3])$. [2 marks]