1.) We want to prove the *NP-hardness* of MULTIPROCESSOR SCHEDULING. Your task is to give a polynomial time reduction from PARTITION (which is *NP-complete*) to MULTIPROCESSOR SCHEDULING. Note that you have to provide only the reduction and not the proof of correctness of the reduction. The definition of these two problems is given below:

**PARTITION:**
Instance: A finite set of *n* positive integers \( S = \{a_1, a_2, \ldots, a_n\} \).
Question: Can the set \( S \) be partitioned into two subsets \( S_1, S_2 \) such that the sum of the numbers in \( S_1 \) equals the sum of the numbers in \( S_2 \)?

**MULTIPROCESSOR SCHEDULING:**
Instance: A set \( J \) of *k* jobs where job \( j_i \) has length \( l_i \), and *m* processors.
Question: Can we schedule all jobs in \( J \) on \( m \) processors such that

a) on each processor the next job in the sequence is started immediately after the preceding job is finished and

b) the total time to execute all jobs on each processor takes the minimum possible time

\[ T_{\text{min}} = \left( \sum_{i=1}^{k} l_i \right) / m . \]

(15 points)

2.) (a) Let \( \varphi^E \) be the following equality logic formula

\[ (x_1 = x_2 \land x_2 = x_3 \land x_3 \neq x_4) \lor x_4 = x_1 . \]

Apply the *Sparse Method* and derive a purely propositional formula \( \varphi^P \), which is equisatisfiable to \( \varphi^E \). Be verbose and tell us which steps you apply, the result obtained in each step and the connection between the different partial results. (5 points)

(b) We consider a simplified variant of Tseitin’s reduction. Let \( \varphi \) be a propositional formula, let \( \Sigma(\varphi) \) be the set of all subformulas of \( \varphi \), and let \( \ell_\varphi \) be the label for \( \varphi \). Prove that

\[ (\land_{\psi \in \Sigma(\varphi)} (\ell_\psi \equiv \psi)) \rightarrow \ell_\varphi \text{ is valid if and only if } \varphi \text{ is valid.} \]

(10 points)

3.) Extend the toy language presented in the course by assert-statements of the form \( \text{assert } e \).

When the condition \( e \) evaluates to true, the program continues, otherwise the program aborts.

Specify the syntax and semantics of the extended language. Determine the weakest precondition, the weakest liberal precondition, the strongest postcondition, and Hoare rules (partial and total correctness) for assert-statements. Show that they are correct.

Treat the assert-statement as a first-class citizen, i.e., do not refer to other program statement in the final result. However, you may use other statements as intermediate steps when deriving the rules. (15 points)

Remember the following properties of wp, wlp, sp, and the Hoare calculus.
4.) Bisimulation

Let \( M_1 = (S_1, I_1, R_1, L_1) \) and \( M_2 = (S_2, I_2, R_2, L_2) \) be two Kripke structures.

Remember, a relation \( H' \subseteq S_1 \times S_2 \) is a bisimulation relation if for each \( (s, s') \in H' \) holds:

- \( L_1(s) = L_2(s') \),
- for each \( (s, t) \in R_1 \) there is a \( (s', t') \in R_2 \) such that \( (t, t') \in H' \), and
- for each \( (s', t') \in R_2 \) there is a \( (s, t) \in R_1 \) such that \( (t, t') \in H' \).

Let \( M_1 = (S_1, \{s_0\}, R_1, L_1) \) and \( M_2 = (S_2, \{t_0\}, R_2, L_2) \) be two (finite) Kripke structures, where \( s_0 \) is the single initial state of \( M_1 \) and \( t_0 \) is the single initial state of \( M_2 \). Consider the following sequence of sets \( P_i \subseteq (S_1 \times S_2) \) and \( N_i \subseteq (S_1 \times S_2) \):

\[
\begin{align*}
N_0 &= \emptyset \\
N_1 &= \{(s, t_0)\} \\
N_{i+1} &= N_i \cup \\
&\quad \{ (s, t) \in P_i \mid (s, t) \notin L_1(s) \neq L_2(t) \} \cup \\
&\quad \{ (s, t) \in P_i \mid \exists (s', t') \in R_1 \land (s, t) \in R_2 \land (s', t') \in N_i \} \cup \\
&\quad \{ (s, t) \in P_i \mid \exists (t, t') \in R_2 \land (s, t) \in R_1 \land (s', t') \in N_i \} \\
P_{i+1} &= P_i \setminus N_{i+1} \cup \text{choose}(\{(s, t) \mid (s, t) \notin P_{i+1} \setminus N_{i+1} \}) \\
&\quad \text{where } \text{choose}(S) \text{ randomly returns an element of the set } S, \text{ if } S \text{ is not empty, and the empty set, otherwise.}
\end{align*}
\]

(a) Show that, for all \( i \in \mathbb{N} \), \( P_i \) and \( N_i \) are disjoint, i.e., \( P_i \cap N_i = \emptyset \). \hspace{1cm} (2 points)

(b) Show that \( N_i \cup P_i \subseteq N_{i+1} \cup P_{i+1} \) holds for all \( i \geq 0 \). \textbf{Hint:} You might use the fact, that \( [(s, t) \in P_j \land (s, t) \notin N_{j+1}] \Rightarrow (s, t) \in P_{j+1} \) holds for all \( j \in \mathbb{N} \). \hspace{1cm} (4 points)

(c) Show that there is a \( n_0 \in \mathbb{N} \) such that \( N_{n_0} = N_n \) and \( P_{n_0} = P_n \) for all \( n > n_0 \). \hspace{1cm} (4 points)

(d) Show that the \( P_{n_0} \) from the preceding question is a bisimulation relation. \hspace{1cm} (5 points)