<iob> .olympiade { fach: “informatik”; bundesland: “brandenburg”; }

10. Informatik-Olympiade vom Land Brandenburg

– Vorrunde –

Hinweise:
Die Aufgaben werden von einzelnen Personen bearbeitet und unter Erbringung der eigenen Leistung.
Die Abgabe (**Abgabetermin = 11.01.2027**) erfolgt (inklusiv der kompletten Namens- und Schul-Nennung) an informatik@blis-brandenburg.de. Häufig kommt es zu Problemen bei dem Empfang von exe-Dateien. In diesem Fall bitte zunächst versuchen, die exe-Datei einfach als txt abzuspeichern mit dem Hinweis, dass es sich um eine exe-Datei handelt. Im Notfall geht auch ein Link zu einem Upload-Ordner von euch, ansonsten schicken wir auf Anfrage einen Link zu einem Upload-Ordner.
Der Abgabe-Ordner soll einen Programm-Ordner mit den kompletten Dateien enthalten, sodass das Programm ausführbar ist, sowie eine erklärende Dokumentation zum Programm (Aufgabe 2) und das Dokument zur Lösung der theoretischen Aufgabe (Aufgabe 1). Bei den Dokumenten sollen möglichst auch visuelle Darstellungen die Erklärung unterstützen.

Auswirkungen der Vorrunde auf den Hauptwettbewerb (Informatik-Olympiade am 27./28.05.2027):
Die beiden Aufgabentypen bereiten euch als Teilnehmer*innen gleichzeitig auf die beiden Olympiaden-Tage vor.

- Die Teilnahme an der Vorrunde erzeugt ein Ranking für die Teilnehmerbewerbung am Hauptwettbewerb, sodass eine vorzeitige Bekanntgabe der persönlichen Teilnahme bei der Olympiade ermöglicht werden kann.
- Die Teilnahme an der Vorrunde ist auch in diesem Jahr weiterhin noch KEINE Voraussetzung zur Teilnahme am Hauptwettbewerb im Juni, allerdings werden die Teilnehmer aus der Vorrunde bevorzugt zur Hauptrunde zugelassen.

Alle Teilnehmer*innen der Vorrunde starten, je nach Platzierungen (Platz 1-3 und dann der Rest), mit unterschiedlichem Punktevorsprung in den Hauptwettbewerb.

Aufgabe 1: Das Funknetz der Forschungsstationen

In einem abgelegenen Forschungsgebiet wurden acht Stationen (**A** bis **H**) errichtet. Die Stationen müssen Daten über Funk miteinander austauschen. Wenn zwei Stationen jedoch zu nah beieinander liegen, überschneidet sich ihr Sendebereich. Senden zwei benachbarte Stationen auf derselben Frequenz (dasselbe Signal), kommt es zu Funkstörungen.
Ein Ingenieurteam hat eine Übersicht derjenigen Stationen erstellt, die sich aufgrund ihrer Nähe gegenseitig stören würden (Konflikte):

Liste der Störverbindungen (Konflikte):

- A stört: B, C, D
- B stört: A, C, D, E
- C stört: A, B, D, E
- D stört: A, B, C, F
- E stört: B, C, F, G
- F stört: D, E, G, H
- G stört: E, F, H
- H stört: F, G



Regeln für die Frequenzvergabe:

1. Jede Station erhält genau einen Frequenzkanal (repräsentiert durch eine Zahl/Farbe: Kanal 1, Kanal 2, etc.).
2. Zwei Stationen, zwischen denen eine Störverbindung besteht, dürfen niemals denselben Frequenzkanal nutzen.
3. Die Gesamtzahl der benötigten Frequenzkanäle soll minimal sein, da jeder zusätzliche Frequenzkanal hohe Lizenzgebühren verursacht.

Aufgabe:

1. Zeichnen Sie den vollständigen Konfliktgraphen des Netzwerks und stellen Sie eine gültige Frequenzzuweisung visuell dar (z. B. durch Einfärben der Knoten oder Kennzeichnen mit Kanalnummern).
2. Begründen Sie mathematisch/logisch (ohne Computerprogramm), warum Ihre Lösung exakt die minimal mögliche Anzahl an Frequenzkanälen verwendet und es unmöglich ist, mit weniger Kanälen auszukommen.
3. Beschreiben Sie ein allgemeines, algorithmisches Verfahren, wie man bei beliebigen Netzwerken schrittweise vorgehen kann, um eine möglichst geringe Anzahl von Frequenzkanälen zu ermitteln.

Aufgabe 2: Das Fracht-U-Boot der Schatzsucher

2.1 Szenario

Ein autonomes Forschungs-U-Boot erkundet ein altes Schiffswrack. Über ein Kamerasystem hat es mehrere wertvolle Artefakte identifiziert. Jedes Artefakt besitzt einen Namen, ein spezifisches Gewicht (in kg) und einen geschätzten historischen Wert (in Forschungspunkten).
Das U-Boot besitzt jedoch zwei physikalische Einschränkungen:

1. **Maximale Traglast (*W*):** Das Gesamtgewicht aller geladenen Artefakte darf den Wert *W* nicht überschreiten.
2. **Begrenzte Greifarm-Kapazität (*k*):** Das U-Boot kann maximal k Artefakte transportieren, unabhängig davon, wie leicht sie sind.

Ziel ist es, eine Auswahl an Artefakten zu treffen, deren Gesamtwert **maximal** ist, ohne die beiden Grenzen *W* und k zu verletzen.

2.2 Aufbau der Eingabe und Ausgabe

Das Programm erhält:

1. Ein Dictionary artefakte mit Name → (Gewicht, Wert)
   Beispiel: {"Arux": (3, 40), "Belo": (4, 50), "Cora": (2, 30), "Drak": (5, 60)}
2. Eine Zahl W= Maximale Traglast in kg.
3. Eine Zahl k= Maximale Anzahl transportierbarer Artefakte.

Ausgegeben werden soll:

- Der maximale Gesamtwert der wählbaren Artefakte.
- Die Liste der ausgewählten Artefakte inklusive ihres Gesamtgewichts und der Gesamtanzahl.
- Wenn kein einziges Artefakt aufgeladen werden kann (z. B. weil jedes Artefakt schwerer als W ist oder k=0), soll ausgegeben werden: "Keine Artefakte transportierbar".

2.3 Beispiel für die Grundaufgabe

Eingabe: artefakte = {"Arux": (3, 40), "Belo": (4, 50), "Cora": (2, 30), "Drak": (5, 60)}; W = 5; k = 2
Ausgabe:
Maximaler Wert: 70
Ausgewählte Artefakte: ['Arux', 'Cora']
Gesamtgewicht: 5 kg (Anzahl: 2)



Erklärung:
Mögliche Kombinationen von max. 2 Artefakten mit Gewicht ≤5:

- ["Arux"] → Gewicht 3, Wert 40
- ["Belo"] → Gewicht 4, Wert 50
- ["Cora"] → Gewicht 2, Wert 30
- ["Drak"] → Gewicht 5, Wert 60
- ["Arux", "Cora"] →
- Gewicht 3+2=5, Wert 40+30=70 (Optimal!)
- *["Belo", "Cora"]* → Gewicht 4+2=6 (> 5, ungültig)

2.4 Aufgabe

a) Schreiben Sie ein Programm, welches für beliebige Eingaben (nach dem Eingabeschema) die optimale Auswahl an Artefakten bestimmt.
b) Ergänzen Sie Ihr Programm so, dass es bei mehreren gleichwertigen optimalen Lösungen alle wertgleichen Kombinationen ausgibt und diejenige hervorhebt, die das geringste Gesamtgewicht besitzt.
c) Erzeugen Sie Beispieldateien zur Eingabe in das Programm mit verschiedenen Schwierigkeitsgraden (inklusive des Falls, dass kein Artefakt mitgenommen werden kann).
d) Schreiben Sie ein Generator-Teilprogramm, das eine Zufallskonfiguration erzeugt (z. B. 30 Artefakte mit zufälligen Werten/Gewichten) und automatisch eine Parameterkombination aus W und k findet, sodass ein exakt vorgegebener Ziel-Gesamtwert V_{ziel} erreicht wird.