



תרגיל בית 4

המתרגל האחראי על התרגיל:

דואר אלקטרוני: osnattal@cs.technion.ac.il

שם: אסנת טל
שעת קבלה: יום ג' 15:00-16:00

שימו לב: בכל הנוגע לתרגיל, נא להפנות שאלות למתרגל המצויין מעלה בלבד.

לוח זמנים:

תאריך הגשה (סופי): 12.1.06, 18:00 אחר הצהריים. לא יתקבלו הגשות באחור.
מועד פרסום התרגיל הבא (משוער): 12.1.06.

אופן ההגשה:

- קראו היטב את דף ההנחיות והמידע המופיע באתר הקורס (תחת הקישור "Syllabus"). שימו לב למלא אחר כל ההוראות המפורטות בו בנוגע להגשת תרגילי בית.
- על קובץ הקוד להיות מתועד, בהתאם לקובץ הדוגמה המופיע באתר הקורס (template.c).
- יש להגיש את תדפיסי התוכניות שכתבתם (עם הפרדה בולטת בין השאלות) בצירוף דף שער ממולא (אותו ניתן להוריד מאתר הקורס). אי שימוש בדף השער יגרור הורדת נקודות.
- ההגשה הינה לתא הקורס אליו אתם רשומים (שימו לב היטב למספר הקורס), בטאוב קומה 1. יש להדק היטב את הדפים המוגשים. נא לא להוסיף ניילונים או תיקיות למיניהן.
- יש להגיש את הקוד שכתבתם גם אלקטרונית, דרך אתר הקורס. את קבצי הקוד יש לכווץ לקובץ zip. אין צורך להגיש קבצים נוספים.
- שימו לב שההגשה האלקטרונית אפשרית רק עד שעת ההגשה המצויינת.
- הגשה באיחור אפשרית רק בתיאום מראש עם המתרגל האחראי על הקורס (רן) – נא לפנות אליו בשעות הקבלה שלו (או בדוא"ל אם זה דחוף) לפי המופיע באתר הקורס. במקרה של הגשה מאוחרת, יש להגיש את התרגיל לתא הדואר של רן בקומה 5 (ולא לתא הקורס).
- שימו לב: אישור על הארכה יינתן רק במקרים חריגים (כפי שמתואר בדף המידע של הקורס). לא יתקבלו פניות כגון "היו לי עוד 3 תרגילים להגיש", "המחשב שלי קרס" או "הכלב אכל לי את הדיסקט".
- תקינות הקלט: יש תמיד לוודא את ערך ההחזרה של פונקציות הקלט. במקרה של כישלון (כלומר scanf() מחזיר מספר קטן מזה שהיה אמור להחזיר, או getchar() מחזיר EOF) יש להדפיס הודעת שגיאה ולהפסיק את ריצת התוכנית.
- בתרגיל זה צורת הדפסת הפלט אינה נוקשה; כלומר, אין הכרח שהפלט שלכם יהיה בדיוק בניסוח ובסדר כמתואר בדוגמאות ההרצה. עם זאת חשוב להקפיד שכל הנתונים שאמורים להופיע בפלט אכן יודפסו (בהתאם למצויין בתאור השאלה), כולל תוצאות חישובים, אינדקסים, הודעות שונות וכו'.



תאור התרגיל

בתרגיל זה נכתוב משחק דמוי Black Jack. במשחק זה יש שני שחקנים, המחשב והמשתתף, ומטרת כל שחקן להגיע לסכום קלפים כמה שיותר קרוב ל-21 (כולל), אך בלא לעבור סכום זה. במשחק יש רק שני תורות: ראשית, תורו של המחשב לשחק עד שהוא מסיים, והתוצאה אליה הוא מגיע נשאר חסויה. לאחר מכן תורו של המשתמש לשחק עד שהוא מסיים. בשלב זה המשחק תם, התוצאה של המחשב מתגלה ומוכרז מי המנצח.

מהלך המשחק:

לכל שחקן תור אחד. ראשית תורו של המחשב, ולאחר מכן תורו של המשתמש. שחקן שמגיע תורו משחק על ידי לקיחת קלפים מהקופה בזה אחר זה עד שהוא מחליט לפרוש, וכל קלף שהוא לוקח, ערכו מצטבר לסכום ערכי הקלפים שלקח עד עכשיו. אם סכום הקלפים של השחקן עובר את הערך 21 (כלומר, 22 ומעלה), אז הוא פורש אוטומטית.

סכום ערכי הקלפים של השחקן בזמן פרישתו הוא התוצאה הסופית שלו במשחק.

מי מנצח במשחק?

- מי שהגיע קרוב יותר ל-21 (כולל), בתנאי שלא עבר את 21. כלומר:
- אם שני השחקים עברו את 21 – יש תיקו, ללא תלות בתוצאות שלהם.
 - אם רק אחד השחקנים עבר את 21, אז השני מנצח.
 - אם אף אחד מהשחקנים לא עבר את 21, מנצח זה שתוצאתו גבוהה יותר (קרובה יותר ל-21). אם לשניהם אותה תוצאה, אז יש תיקו.

אסטרטגית המחשב

המחשב מחליט כל פעם האם לקחת קלף נוסף או לא לפי אסטרטגית המשחק שלו. אסטרטגית המשחק של המחשב נקבעת על ידי המשתמש בתחילת ריצת התוכנית (והמחשב משתמש באותה האסטרטגיה לאורך כל המשחק). האסטרטגיות האפשריות הן:

1. אם סכום הקלפים של המחשב קטן או שווה ל-15, הוא מחליט לקחת קלף נוסף. אם סכום קלפיו 16 או יותר, הוא פורש.
2. המחשב תמיד מחליט לקחת קלף נוסף בהסתברות של 0.7, ובהסתברות 0.3 בוחר לפרוש.
3. נניח התוצאה הנוכחית של המחשב היא x , אז המחשב לוקח קלף נוסף בסיכוי $(21-x)/21$. לדוגמה, אם התוצאה הנוכחית של המחשב היא 14, אז המחשב יקח קלף נוסף בהסתברות $7/21=1/3$, ואם התוצאה היא 3, אז הוא יקח קלף נוסף בהסתברות $18/21=6/7$.

על מנת להגריל האם לקחת קלף נוסף בהסתברות מסוימת, תוכלו להשתמש בפונקציה `takeAnotherCard()` המסופקת בהמשך.

הקלות למשתמש

בניגוד למחשב, בכל פעם שהשחקן מחליט לקחת קלף נוסף ניתנים לו מספר נסיונות לעשות כן. כלומר, במידה והוא איננו מרוצה מהקלף שקיבל, הוא יכול לבקש קלף אחר במקומו. מספר הנסיונות המותרים נקבע בתחילת המשחק על ידי המשתמש, והוא יכול להיות בין 1 ל-5. שימו לב שאם הוא בוחר בנסיון 1 בלבד, אז לשחקן אין כל אפשרות לבקש קלף אחר אם אינו מרוצה מהקלף שקיבל. אם הוא בוחר ב-2 נסיונות אז יש לו הזדמנות אחת לבקש קלף אחר, ואם הוא בוחר ב-5 נסיונות מותר לו עד 4 פעמים לבקש להחליף את הקלף שקיבל. כמובן שאם המשתמש מנצל את כל הנסיונות שלו להחליף קלף, הקלף אחרון שניתן לו מחייב אותו ואין אפשרות לשנותו.



דוגמאות למשחקים:

דוגמא א':

- נניח מספר הנסיונות של המשתמש הוא 3. המחשב שיחק, קיבל סה"כ 24 אולם המשתמש אינו יודע זאת. כעת תורו של המשתמש.
1. המשתמש מקבל 8. הוא נשאל אם הוא רוצה לנסות להחליף את הקלף, והמשתמש עונה "לא".
 2. המשתמש מקבל 9. הוא נשאל אם הוא רוצה לנסות להחליף את הקלף, והמשתמש עונה "לא". לכן הוא מקבל את ה-9, והתוצאה שלו עד כה היא 17.
 3. המשתמש מקבל 8, מה שמביא אותו לעבור את 21. לכן הוא מבקש להחליף את הקלף. כעת הוא מקבל 6, שגם כן מביא אותו לעבור את 21. הוא מבקש שוב להחליף את הקלף (זו הפעם האחרונה שמוותר לו לעשות כן כיוון שיש לו 3 נסיונות). לרוע מזלו הוא מקבל 10, ולכן תוצאתו היא 27. כיוון שהוא עבר את 21, התור שלו מסתיים.
 4. כעת תוצאת המחשב מתגלה – למשתמש היה מזל כיוון שהמחשב גם כן עבר את 21 ולכן המשחק מסתיים בתיקו.

דוגמא ב':

- נניח מספר הנסיונות של המשתמש הוא 1. המחשב שיחק וקיבל 19.
1. המשתמש לוקח קלף מהקופה, ומקבל 7.
 2. המשתמש לוקח קלף ומקבל 9. כעת תוצאתו היא 16.
 3. המשתמש מחליט שלא לקחת קלף נוסף מהקופה. המשחק מסתיים בנצחונם של המחשב.

דוגמא ג':

- נניח מספר הנסיונות של המשתמש הוא 2. המחשב שיחק וקיבל 17.
1. המשתמש לוקח קלף מהקופה, ומקבל 4. הוא נשאל אם הוא רוצה לנסות שוב, ועונה "לא".
 2. המשתמש לוקח קלף נוסף ומקבל 9. המשתמש מבקש להחליף את הקלף ומקבל 8. כיוון שיש לו רק 2 נסיונות, הוא חייב להישאר עם ה-8 ותוצאתו כעת היא 10.
 3. המשתמש לוקח קלף נוסף ומקבל 7. הוא נשאל אם הוא רוצה להחליף, ועונה "לא". לכן תוצאתו היא כעת 17.
 4. המשתמש מחליט שלא לקחת קלף נוסף מהקופה, ולכן המשחק מסתיים בתיקו.

הערות

- הקלפים במשחק ממוספרים מ-1 עד 13. הסיכוי לקבלת כל אחד מהקלפים הוא תמיד זהה, ללא תלות בקלפים שנתקבלו עד כה.
- יש לכתוב את הקוד ע"פ עקרונות top-down design וחלוקה הגיונית של הקוד לפונקציות.
- יש לבדוק תמיד את הצלחת ונכונות הקלט, ובמקרה של כישלון טפלו בכך על פי בחירתכם.
- ניתן בשאלה זו להשתמש בפונקציה `exit()` במידת הצורך.
- זכרו לקרוא בתחילת התכנית ל `(srand(time(0)))` ע"מ שפונקציית `rand()` תחזיר ערכים שונים בריצות שונות של התכנית.
- הגדירו את הפרמטרים הבאים של המשחק כקבועי `#define`: את הערך 21, את ערך הקלף המקסימלי 13, את הערך 15 באסטרטגיה מס' 1, את ההסתברות 0.7 באסטרטגיה מס' 2, ואת מספר הנסיונות המקסימלי שלמשתמש מותר לבחור בו בתחילת המשחק (5).
- בתרגיל זה יינתן בonus של נקודה עבור תכנון טוב של הקוד וחלוקתו לפונקציות ע"פ עקרונות top-down design.



מצורף קוד הפונקציה `takeAnotherCard()`, שמקבלת כפרמטר הסתברות ממשית p . הפונקציה מחזירה את הערך 1 (כלומר לקחת קלף) בהסתברות p , ואת הערך 0 (כלומר לא לקחת קלף) בהסתברות $1-p$. אם $p > 1$ היא תמיד מחזירה 1, ואם $p < 0$ היא תמיד מחזירה 0.

```
int takeAnotherCard(double p)
{
    if (p >= 1) return 1;
    if (p <= 0) return 0;

    return ((rand() / ((double)RAND_MAX)) <= p);
}
```

ריצות לדוגמה

ריצה א':

```
Welcome to the BLACK-JACK game!
```

```
-----
```

```
Please enter computer strategy (1,2 or 3): 1
```

```
Please enter number of user tries (1-5): 3
```

```
Computer has played. Your turn!
```

```
Your score is 0.
```

```
Would you like another turn? (y/n): y
```

```
Card is: 12. You have 2 tries left. Would you like a different card? (y/n): n
```

```
Your score is 12.
```

```
Would you like another turn? (y/n): y
```

```
Card is: 8. You have 2 tries left. Would you like a different card? (y/n): y
```

```
Card is: 13. You have 1 tries left. Would you like a different card? (y/n): y
```

```
Card is: 7.
```

```
Your score is 19.
```

```
Would you like another turn? (y/n): n
```

```
Your score: 19, computer's score: 22
```

```
You have won!
```



ריצה ב':

```
Welcome to the BLACK-JACK game!
-----

Please enter computer strategy (1,2 or 3): 3
Please enter number of user tries (1-5): 1
Computer has played. Your turn!

Your score is 0.
Would you like another turn? (y/n): y
Card is: 13.

Your score is 13.
Would you like another turn? (y/n): y
Card is: 10.

Your score: 23, computer's score: 8
You have lost...
```

ריצה ג':

```
Welcome to the BLACK-JACK game!
-----

Please enter computer strategy (1,2 or 3): 2
Please enter number of user tries (1-5): 2
Computer has played. Your turn!

Your score is 0.
Would you like another turn? (y/n): y
Card is: 6. You have 1 tries left. Would you like a different card? (y/n): y
Card is: 10.

Your score is 10.
Would you like another turn? (y/n): y
Card is: 9. You have 1 tries left. Would you like a different card? (y/n): n

Your score is 19.
Would you like another turn? (y/n): n

Your score: 19, computer's score: 10
You have won!
```

בהצלחה!