

99

מחברת מס' 1
מתוך 1 מחברות

86

הוראות לנבחנים ולנבחנות (נכתבו בלשון זכר אך נועדו לשני המינים)
לפני התחלת הבחינה סלא את כל הפרטים הבאים בכתב ברור וקרא בעיון את ההוראות:

הפקולטה למדעים מדויקים ע"ש ריימונד ובברלי סאקלר

- הנך נדרש לשמור על טוהר הבחינה ועל עבודה עצמית ולהישמע להוראות המשגיחים ולנוהלי האוניברסיטה. אין להעתיק, אין לדבר ואין להעביר חומר בין הנבחנים.

נבחן הנוהג בניגוד להוראות צפוי להפסקת בחינתו ולהעסדה לדין משפטי.

- על הנבחן להבחן בחדר שבו הוא רשום.
- אין להחזיק טלפונים ניידים או אמצעי תקשורת ומכשירים אלקטרוניים כלשהם בזמן הבחינה. על הנבחן להניח את כל חפציו האישיים בצד החדר הרחק מסקום מושבו.
- אין להחזיק בהישג יד, בחדר הבחינה או בסמוך לו, כל חומר הקשור לבחינה או לקורס פרט לחומר שהשימוש בו הותר בכתב על ידי המורה.
- קריאת השאלון מותרת רק לאחר קבלת רשות מהמשגיח.
- נבחן לא יעזוב את מקומו ולא את חדר הבחינה בטרם סיים את הבחינה ללא קבלת רשות מהמשגיח. בעת יציאה מן החדר, יפקיד הנבחן את מחברות הבחינה והשאלון (טופס הבחינה) בידי המשגיח.
- נבחן שנכנס לחדר הבחינה וקיבל את השאלון לידיו, לא יחא רשאי לעזוב אותו אלא כעבור חצי שעה לפחות סמועד תחילתה ורק לאחר שיחזיר למשגיח את המחברת ואת השאלון, ויקבל מסנו את התעודה המזהה שאותה מסר עם כניסתו לכיתה. נבחן שהחליט לעזוב בלי לכתוב את הבחינה ייחשב כמי שנבחן במועד זה וציונו יהיה "0".
- אין לכתוב את השם או כל פרט מזהה אחר בתוך המחברת. פרטי הנבחן ימולאו על כריכת המחברת במקום המיועד לכך בלבד.
- אין לתלוש דפים מהמחברת. טיוטה תיכתב בתוך המחברת בלבד. אין להשתמש בדפים שהביא הנבחן.
- יש לכתוב את התשובות בעט כחול או שחור, בכתב יד ברור ונקי. בתום הבחינה יחזיר הנבחן את המחברת והשאלון ויקבל סיד המשגיח את התעודה המזהה.

11. אין לכתוב מעבר לקו האדום משני צידי הדף.

בהצלחה.

תאריך הבחינה 28.1
שם הקורס פרויקט תוכנה
שם המורה אמיר קרן

מס' זיהוי
(העתק מכרטיס הנבחן/התלמיד)
3 | 0 | 5 | 7 | 4 | 8 | 0 | 0 | 6



לשימוש המועדף והבוחן:

הציון _____
המחברת נבדקה ביום _____
חתימת המורה _____

4439144

35 | 1
34 | 2
30 | K

YD 38A

סמסטר א' תשע"ו, מועד א'

28/01/2016

משך הבחינה: שעתיים

חומר עזר: שני דפי עזר

בחינה בקורס: פרויקט תוכנה

אמיר קירש, מואב ערר

הנחיות כלליות לבחינה:

- בראש העמוד הראשון של טופס המבחן (עמוד זה) יש לציין את מספר תעודת הזהות.
- בבחינה שאלות פתוחות ואמריקאיות בעלות ניקוד משתנה, בסך של 100 נק'.
- את התשובות לשאלות 3-5 יש למלא בטבלה המיועדת לכך (בעמוד זה).
- במידת הצורך יש לתעד את התשובה לשאלות הפתוחות (אפשר בעברית).
- אין צורך להציג #include וכן אין צורך להציג הפרדה לקבצי h ו-c.

בהצלחה !

טבלת תשובות לשאלות 3-5

יש לסמן X בעמודה המתאימה בלבד

א	ב	ג	ד	ה	ו
א		ג	ד		
		ג			
					ו

שאלה 3

שאלה 4

שאלה 5

נימוק לתשובות 3-5 (חובה לספק נימוק במידה ונבחרה תשובה ו' !!!)

לא חובה לספק נימוק עבור תשובות א' עד ה'

- תשובה נכונה ללא נימוק תזכה בנקודות (פרט לתשובה ו' שמחייבת נימוק!)
- כאשר ישנה תשובה נכונה וזאת לא סומנה, הנימוק גם אם הוא נכון, לא יזכה בנקודות

שאלה 3 $\frac{d}{dt} \int_{\Omega} u^2 dx = -2 \int_{\Omega} u \Delta u dx = 0$

והתוקף הרגיש ס (עין ברירה מחול) ואם צג מסכר צבא וצק צוץ

שאלה 4 גידן ציון גם כי חזן של א' גלובל יה' 3

שאלה 5 זמן 3 דקות 6 אמות 15 נקודות

ל"ב 13 כ' א' 11/12 תרס"א

כ"ק וצ"ק ע"ג 1313

per 18 miles per hour is 178 ft per second

1313

שאלה מס' 1 (35 נק')

כתבו תוכנית המקבלת ב-command line:

(א) פרמטר ראשון: שם קובץ

(ב) אוסף מחרוזות

על התוכנית לחשב את מספר המופעים של כל תו בכל המחרוזות שמופיעות מיד לאחר שם הקובץ (לא כולל שם הקובץ!) ולהדפיס לקובץ גרף של סולמיות לפי מספר המופעים של התווים, בסדר ה-Ascii שלהם. ניתן להניח שהתווים יכללו אך ורק תווים פשוטים שניתן להדפיס מתוך טווח ה-Ascii (לא כולל למשל תו של ירידת שורה, טאב, וכדומה).

התוכנית רוצה לקבל שם של קובץ ליצירה בלבד (לא לדרוס קובץ קיים). במידה ולא נמסרו פרמטרים בכלל, יש להוציא הודעה מתאימה ולצאת. במידה ונמסר שם קובץ בלבד ללא מחרוזות יודפס גרף ריק לקובץ (כלומר, כלום) – וזה חוקי. במידה ולא ניתן לייצר את הקובץ מכל סיבה (למשל כי הוא כבר קיים), התוכנית תוציא הודעה ותצא, יש להודיע שיש בעיה ביצירת הקובץ אבל אין צורך לפרט בהודעה מה הבעיה.

דוגמה:

עבור הקלט הבא (הנח שקראנו לתוכנית DrawFrequencyChart ואנחנו מריצים אותה ב-Linux):

./DrawFrequencyChart out.txt aaAA B ### AA

יודפס לקובץ out.txt התוכן הבא:

```
# ###
A ####
B #
a ##
```

יש לדאוג שהתוכנית שתיכתב תהיה בנויה מפונקציות פשוטות ונכונות (ולא לכתוב הכל ב-main למשל).

שאלה מס' 2 (35 נק')סעיף א': (10 נק')

כתבו את הפונקציה:

```
size_t count (const void* arr[], size_t size, bool (*func) (const void*))
```

הפונקציה מקבלת את הפרמטרים הבאים:

- מערך בשם arr של איברים מסוג const void* (פוינטר const כלשהו)
- את גודל המערך באמצעות size
- פונקציית func שמקבלת const void* ומחזירה bool

הפונקציה count תפעיל את הפונקציה func על כל איבר במערך שקיבלה ותחזיר את מספר האיברים שהפונקציה func החזירה עבורם true.

סעיף ב': (20 נק')

כתבו את הפונקציה:

```
bool allCapital(const void* s)
```

הפונקציה תקבל מחרוזת מסוג const char* "בתחפושת" של const void* ותבדוק האם המחרוזת מכילה רק תווים שהם אותיות גדולות באנגלית (בטווח 'A' עד 'Z').

בנוסף, כתבו את הפונקציה:

```
size_t countAllCapital(const char* arr[], size_t size)
```

הפונקציה מקבלת את הפרמטרים הבאים:

- מערך בשם arr של איברים מסוג const char*
- את גודל המערך באמצעות size

על הפונקציה להשתמש בפונקציה שבסעיף א' (גם אם לא כתבת אותה) ובפונקציית העזר שכתבת בסעיף זה, על-מנת לספור כמה מחרוזות במערך מכילות רק תווים שהם אותיות גדולות באנגלית.

סעיף ג': (5 נק')

כתבו main שיפעיל את הפונקציה countAllCapital שכתבת בסעיף ב' ויקבל חזרה מהפונקציה את המספר 1.

שאלה מס' 3 (10 נק')

נתונה התכנית הבאה:

```
1. void swap(const char** ptr_s1, const char** ptr_s2) {
2.     const char* temp = *ptr_s1;
3.     *ptr_s1 = *ptr_s2;
4.     *ptr_s2 = temp;
5. }
6.
7. int main() {
8.     const char* s1 = "Hello";
9.     const char* s2 = "world";
10.    swap(&s1, &s2);
11.    printf("s1: %s, s2: %s", s1, s2);
12. }
```

מה תדפיס שורה 11?

התוכנית לא תעבור קומפילציה כי לא ניתן להחליף מחרוזות const-יות. ☒ א

התוכנית תיכשל בלינקינג כי הפרמטרים שנשלחים לפונקציה בשורה 10 אינם מתאימים. ☒ ב

התוכנית תעוף בזמן ריצה על ניסיון לגשת לזיכרון שלא הוקצה. ☒ ג

התוכנית תדפיס: s1: world, s2: Hello ☒ ד

התוכנית תדפיס: s1: Hello, s2: world ☒ ה

אף תשובה אינה נכונה. ☒ ו

שאלה מס' 4 (10 נק')

נתונה התכנית הבאה:

```
1. struct A {
2.     int i;
3. } a;
4.
5. void doIt(struct A a) {
6.     a.i = 3;
7. }
8.
9. void print(struct A a) {
10.    printf("%d", a.i);
11. }
12.
13. int main() {
14.    doIt(a);
15.    print(a);
16. }
```

לירן, טירן, שירן ומורן התכוננו למבחן וטענו טענות.

לירן טען: התכנית תדפיס מספר זבלי כלשהו.

טירן טען: התכנית תדפיס 3.

✓ שירן טען: אם נשנה בשורה 5 את שם הפרמטר ל-b, מבלי לגעת בשאר התכנית, התכנית תדפיס 3.

מורן טען: אם נשנה בשורה 9 את שם הפרמטר ל-b, מבלי לגעת בשאר התכנית, התוכנית תדפיס 3.

מי מהחבורה צודק?

א. טירן בלבד!

ב. לירן, שירן ומורן

ג. לירן ושירן

ד. לירן ומורן

ה. לירן בלבד!

ו. אף תשובה אינה נכונה.

שאלה מס' 5 (10 נק')

נתונה התכנית הבאה:

```
1. int main() {  
2.     int i = 3;  
3.     int* ptr1 = &i;  
4.     int* ptr2 = (int*)malloc(sizeof(int));  
5.     ptr2 = ptr1;  
6.     *ptr2 = 15;  
7.     printf("%d\n", i);  
8.     free(ptr2);  
9. }
```

לירן, טירן, שירן ומורן התכוננו למבחן וטענו טענות.

לירן טען: שורה 7 תדפיס: 3. ✗

טירן טען: שורה 7 תדפיס: 15. ✓

שירן טען: בשורה 5 אנחנו משאירים זיכרון צף, כלומר זיכרון שהוקצה ולעולם לא ישוחרר. ✓

מורן טען: ה-free שבשורה 8 יגרום לתכנית לעוף. ✗

מי מהחבורה צודק?

א. לירן בלבד!

ב. טירן בלבד!

ג. לירן ושירן

ד. טירן ומורן

ה. לירן, שירן ומורן

ו. אף תשובה אינה נכונה. ✓

בהצלחה!


```
#define ASCII_COUNT=256
main(int argc, char* argv){
```

```
if (fout == NULL){
    return;
}
FILE* fout = check_parameters(argc, argv);
size_t count = parse_input(argc, argv);
```

```
print_to_file(count, fout);
free(count);
fclose(fout);
```

```
File* check_parameters(int argc, char* argv){
```

```
if (argc == 1) { // no parameter received
    printf("please provide parameters");
    return NULL;
}
```

```
FILE* fout = fopen(argv[1], "r");
```

```
if (fout == NULL){
    printf("can't creat file");
}
```

```
return fout;
```

```
}
```

```
size_t parse_input(argc, argv){
```

```
size_t count = (int*) malloc(sizeof(int) * ASCII_COUNT);
```

// run over all words, skip file output and program name

```
for (size_t i = 2; i < argc; i++) {
```

// run over all chars in word

```
for (size_t j = 0; argv[i][j] != '\0'; j++) {
```

```
count[argv[i][j]]++; // increase this char count
```

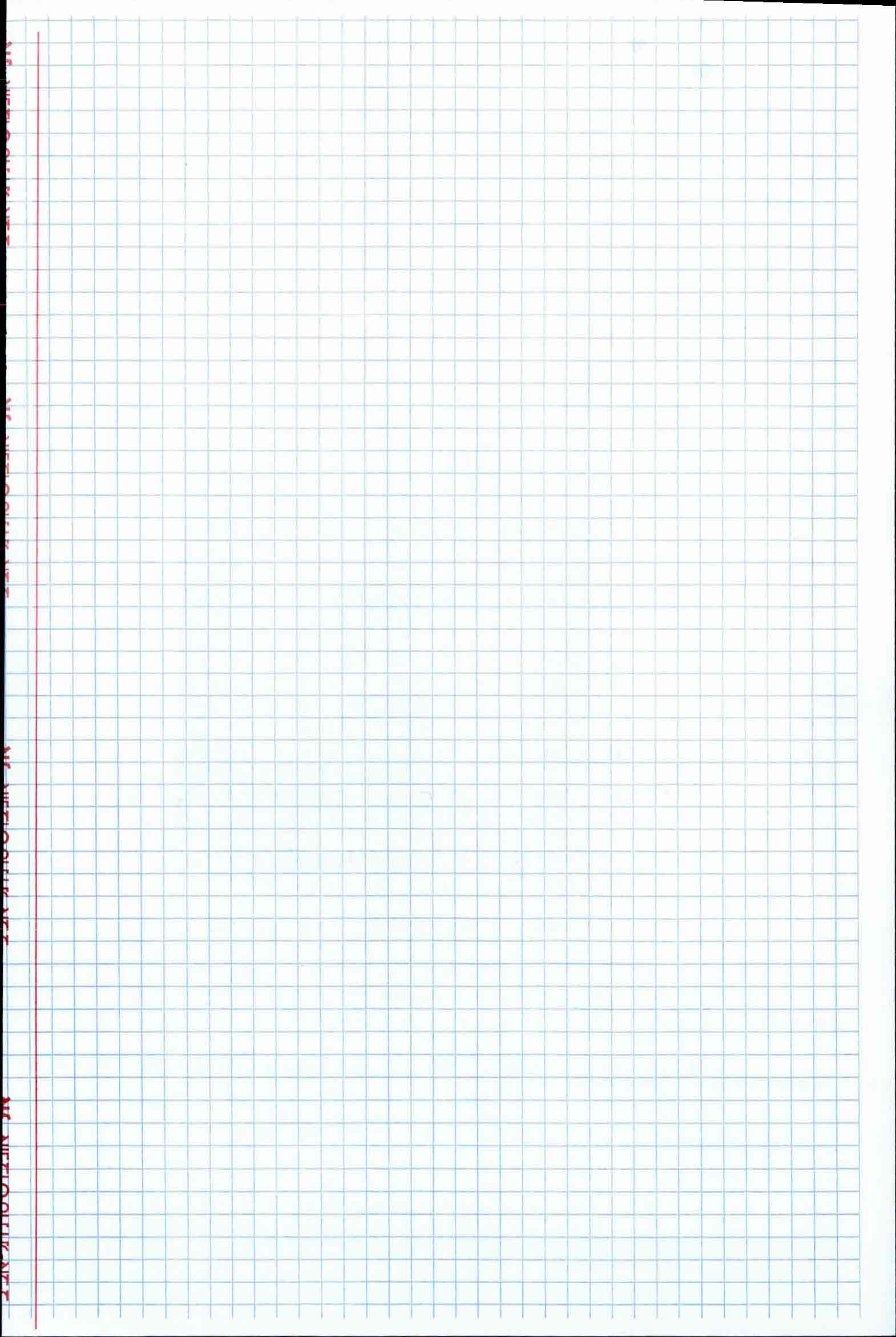
```
}
```

```
}
```

```
return count;
```

int le nagh 12
: 12 nagh 12
(int*) argv[i][j]

12 12 12




```

void print_to_file (size_t* count, FILE* fout) {
    for (size_t i=0; i < ASCII_COUNT; i++) {
        if (count[i] != 0) {

```

```

            fprintf(fout, "%d", count[i]);
            for (size_t j=0; j < count[i]; j++)
            fprintf(fout, "%d", count[i]);

```

```

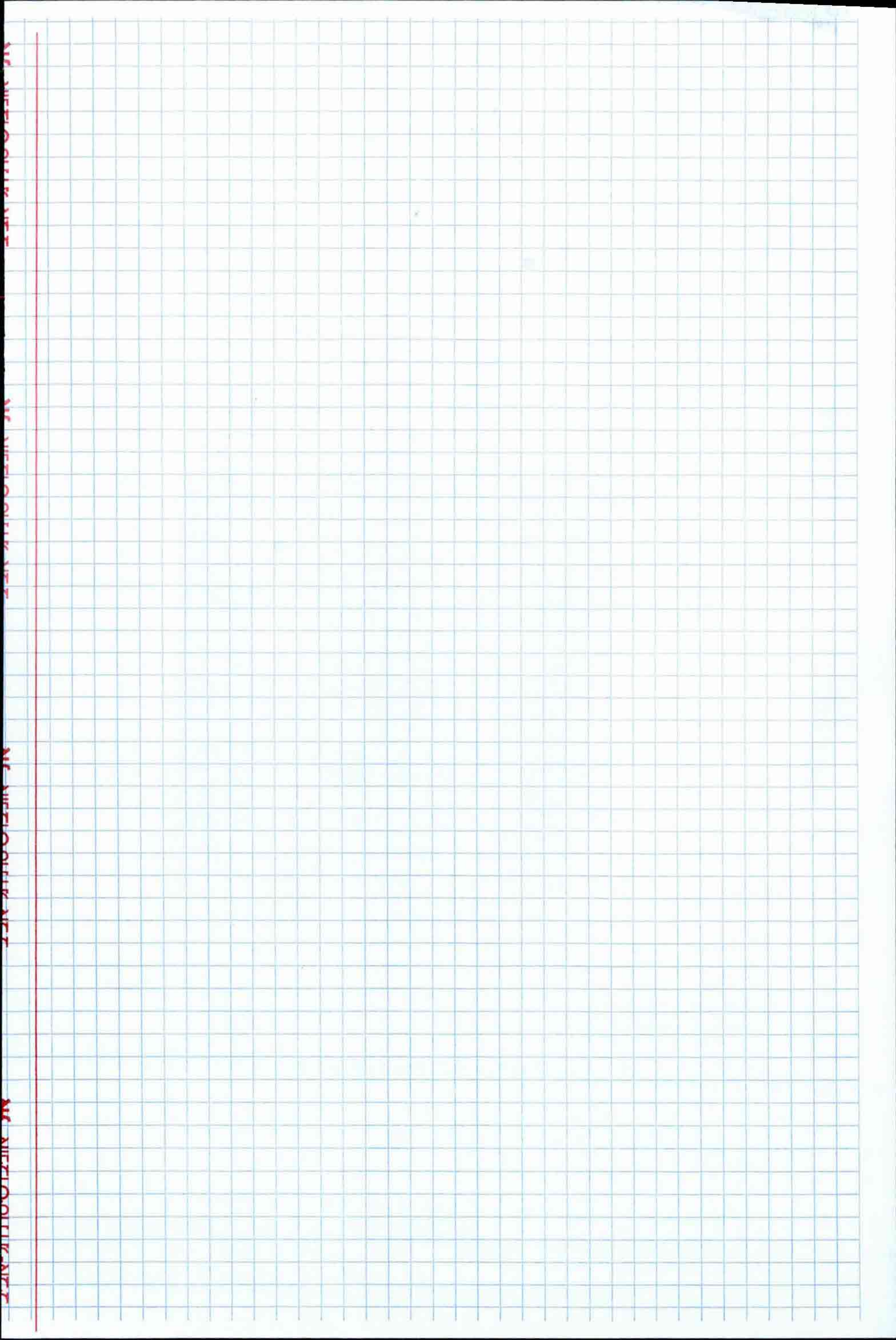
            fprintf(fout, "%c ", (char*)i); // print the Ascii
            for (size_t j=0; j < count[i]; j++) { // value of i
                fprintf(fout, "#");
            }
            fprintf(fout, "\n");

```

```

        }
    }
}

```

size_t count(const void* arr[], size_t size, bool (*func)(const void*)) { .10 ②

```
size_t count = 0;
for (size_t i = 0; i < size; i++) {
    if (func(arr[i])) {
        count++;
    }
}
return count;
```

③

bool allCapital(const void *s) {

```
for (size_t i = 0; i < size; i++) {
    (char*) s; // cast to char*
    for (size_t j = 0; s[j] != '\0'; j++) {
        if (s[j] < 'A' || s[j] > 'Z') {
            return false;
        }
    }
}
```

```
return true;
```

size_t countAllCapital(const char* arr[], size_t size) {

```
size_t count = 0;
for (size_t i = 0; i < size; i++) {
    if (allCapital((void*) arr[i])) {
        count++;
}
```

```
return count((void**) arr, size, allCapital);
```

int main() {

```
const char* arr[] = {"CAPITAL", "lower"};
```

```
size_t res = countAllCapital(arr, sizeOf(arr));
```

```
printf("res = %lu", res);
return 0;
```

}

size of(arr) / size of(arr[0])

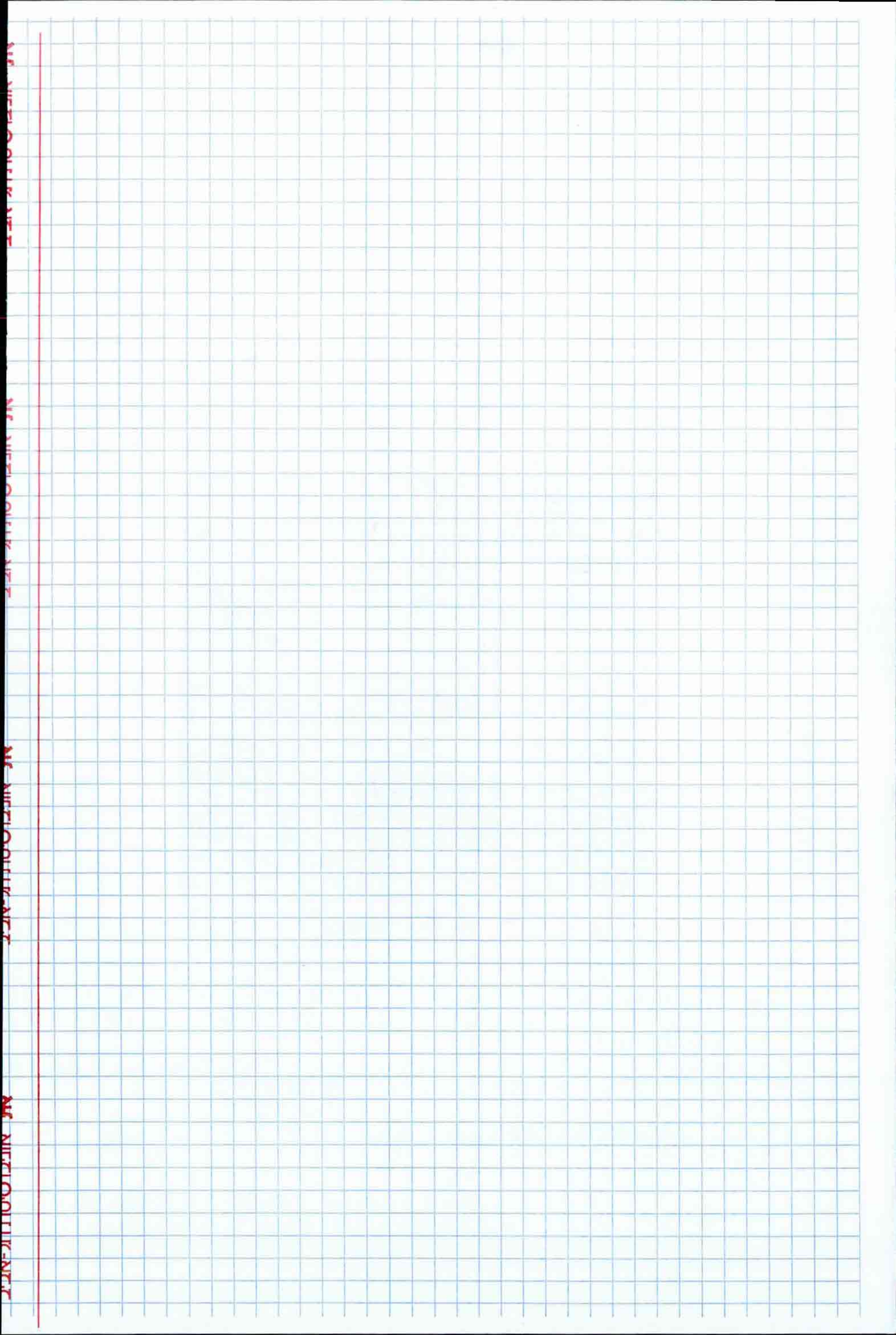
casting
res
size of(arr)
size of(arr[0])

②

③

④

: P2 10/6



NY NECTON LIC-NT-2

NY NECTON LIC-NT-2

NY NECTON LIC-NT-2

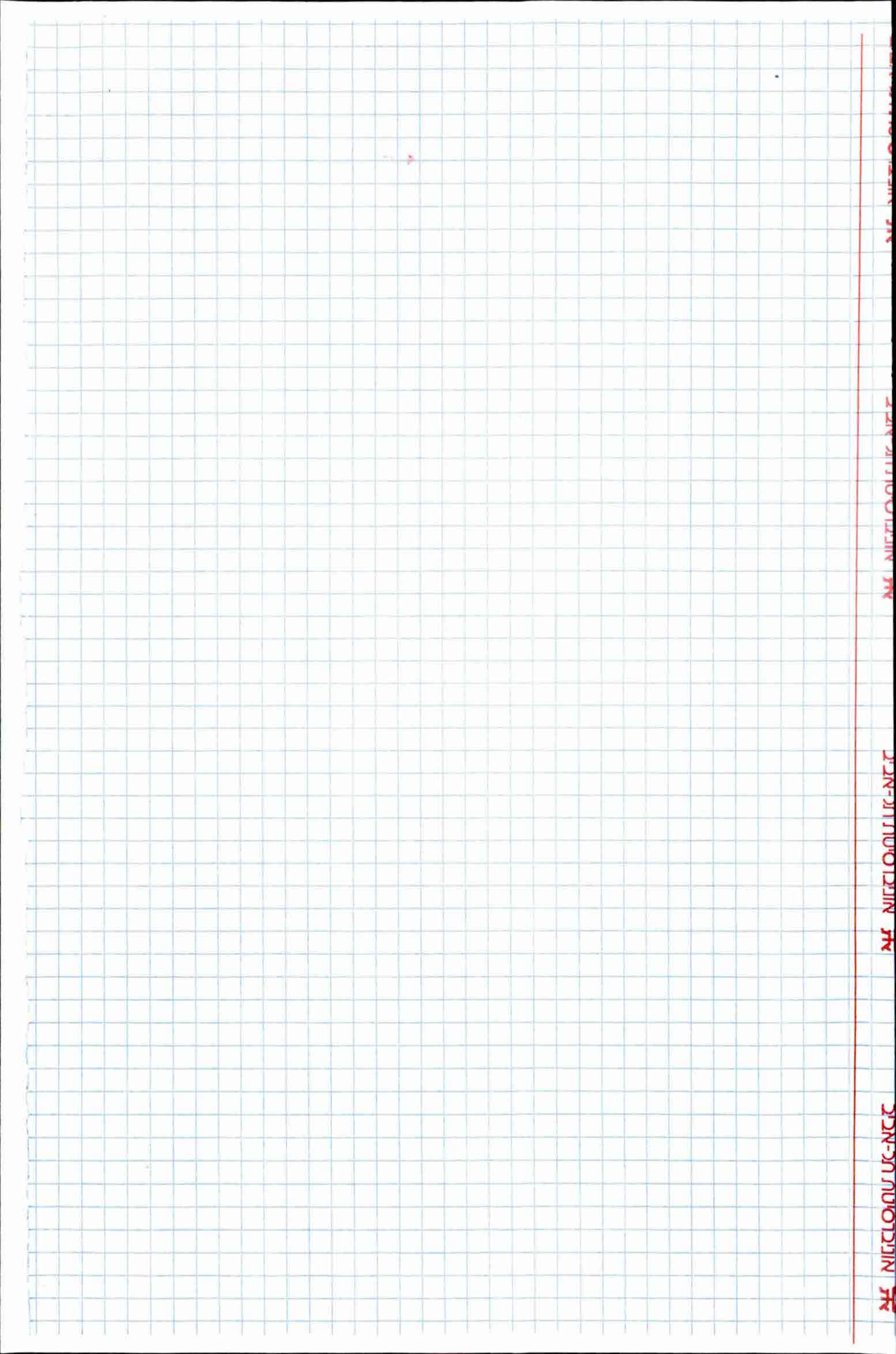
NY NECTON LIC-NT-2

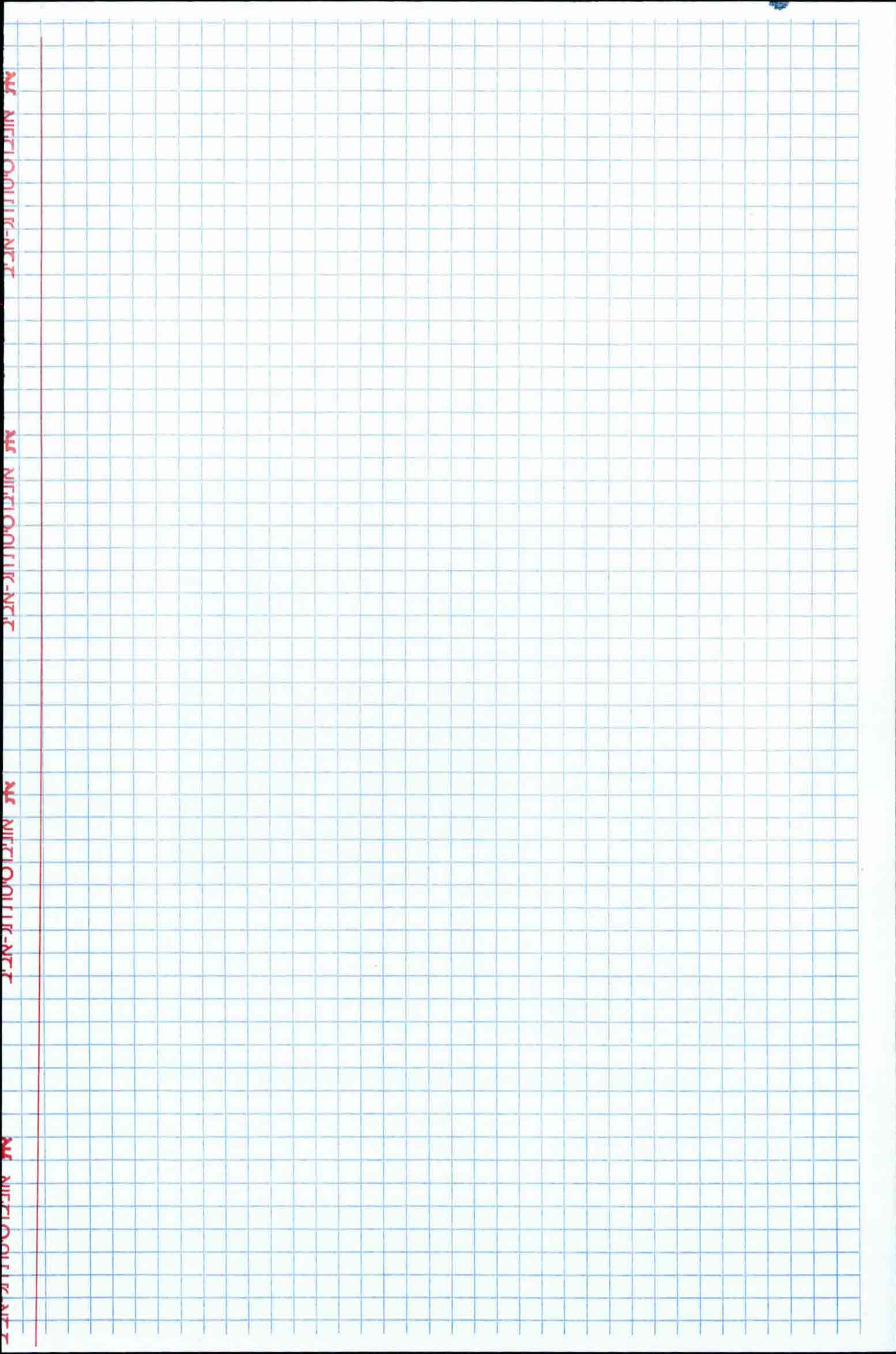
NY NICTD OULIIC-NCT

NY NICTD OULIIC-NCT

NY NICTD OULIIC-NCT

NY NICTD OULIIC-NCT







לשימוש המרצה בלבד

טבלה לחישוב ציונים

[illegible]