

סמסטר א' תשס"ד

מועד: א' 13/2/2004

משך הבחינה: 3 שעות

חומר עזר: שני דפי עזר

בחינה בקורס: תוכנה 1  
מרצה: פרופ' דניאל כהן-אור וגב' איריס רוזנבלום

### הנחיות כלליות לבחינה:

- המבחן מורכב משלוש שאלות בעלות ניקוד לא אחיד.
- חובה לתעד את התשובות. (כמובן מותר בעברית).
- נא לכתוב בכתב קריא ולא מחובר.

### בהצלחה!

#### 1. פלינדרומים (30 נקודות, כל סעיף 15):

מחרוזת נקראת פלינדרום אם אנו מקבלים אותה תוצאה בקריאה מימין לשמאל ומשמאל לימין.  
( למשל: "abcdba" ).

1. כתוב פונקציה בשם reverse המקבלת מחרוזת והופכת אותה. לדוגמה את asdf23v היא תהפוך ל-v32fdsa.
2. כתוב פונקציה המקבלת מחרוזת ומחזירה את אורך תת המחרוזת הארוכה ביותר שהיא היא פלינדרום.
- לדוגמא: עבור המחרוזת "madam, I'm adam!" הפונקציה תחזיר 5.
- שים לב, הפונקציה מבדילה בין אותיות קטנות לגדולות.
- נתון שאורך המקסימאלי של פלינדרום הינו 6 תווים.

#### 2. מטריצות דלילות (30 נקודות, כל סעיף 15):

נגדיר יצוג למטריצות דלילות ע"י הטיפוסים הבאים:

```
typedef struct _Entry{
    int pos;
    double value;
    struct _Entry *next;
}Entry;
```

```
typedef struct _Row{
    int index;
    Entry *head;
}Row; struct _Row *next;
```

```
typedef struct{
    int rows, columns;
    Row *head;
}SparseMatrix;
```

הטיפוס SparseMatrix, המייצג מטריצה ממשית דלילה, מכיל את מספר השורות והעמודות במטריצה, וכן מצביע לרשימה מקושרת בה כל שורה במטריצה מיוצגת ע"י תא (מטיפוס Row), המכיל את אינדקס השורה וכן מצביע לרשימה מקושרת המכילה את כל האיברים השונים מ-0 בשורה זו. כל איבר כזה מיוצג ע"י תא (מטיפוס Entry) המחזיק את מיקומו בשורה ואת ערכו. רשימה זו ממוינת לפי המיקום בשורה, מקטן לגדול.

א. ממש את הפונקציה הבאה:

```
SparseMatrix *create(int **mat, int rows, int cols);
```

אשר מקבלת מטריצה כפרמטר (כולל גודלה) ומחזירה את המטריצה ביצוג הנ"ל.

ב. ממש (ביעילות!) את הפונקציה הבאה:

```
SparseMatrix *add(SparseMatrix *m1, SparseMatrix *m2);
```

אשר מקבלת כפרמטרים שתי מטריצות בעלות גודל זהה ביצוג שהוגדר ומחזירה מטריצה (חדשה) שהיא סכומן.

3. הפיכת ביטים (40 נקודות, כל סעיף 20):

א. כתוב פונקציה המקבלת מצביע ל-char והופכת את היצוג הבינארי שלו.

למשל, אם ch הוא תו char המכיל 01100010, אחרי הקריאה:

```
ReverseBits(&ch)
```

ch יכיל 01000110.

ב. כתוב תוכנית המקבלת כפרמטר שם קובץ בינארי המכיל 99 תווים ויוצרת קובץ טקסט חדש עם אותו שם אבל סיומת r.

לדוגמה אם שם קובץ הבינארי הוא sam2 אזי שם קובץ הטקסט יהיה sam2.r.

בקובץ הטקסט יש לכתוב את כל 99 המחרוזות המתארות את התו ההפוך מסעיף א.

45

מחברת מס' \_\_\_\_\_  
מתוך \_\_\_\_\_ מחברות

אוניברסיטת תל-אביב  TEL AVIV UNIVERSITY

**לפני התחלת הבחינה מלא את כל הפרטים הבאים בכתב ברור  
וקרא בעיון את ההוראות:**

1. על הנבחן להיבחן רק בחדר שבו הוא רשום.
  2. עם הכניסה לחדר הבחינה יש להניח את החפצים בצד לדבות מכשיר קשר ואמצעי תקשורת אחרים כשהם כבויים.
  3. אסור להחזיק בהישג יד, בחדר הבחינה או בסמוך לו, כל חומר הקשור לבחינה/לקורס פרט לחומר שהשימוש בו הותר בכתב על ידי המורה.
  4. יש למלא את הפרטים על מחברת הבחינה במקום המיועד לכך בלבד. אין לכתוב את השם או כל פרט מזהה אחר בתוך המחברת.
  5. יש להישמע להוראות המשנית. נבחן לא יעזוב את **29/30:1** מקומו ללא קבלת רשות המשנית. הפונה בשאלה **30/30:2** או בבקשה ירים את ידו. **36/40** :3
- לשימוש המורה הבוחן:
- |                   |    |
|-------------------|----|
| הציון             | 95 |
| המחברת נבדקה ביום |    |
| חתימת המורה       |    |
6. נבחן שנכנס לחדר הבחינה וקיבל את השאלון (טופס הבחינה) לידו ייחשב כמי שנבחן במועד זה. היה והחליט לא לכתוב את הבחינה, לא יהא רשאי לעזוב את חדר הבחינה, אלא כעבור חצי שעה ממועד תחילתה ולאחר שהחזיר את המחברת והשאלון. ציונו בבחינה יהיה "ס".
  7. קריאת השאלון מותרת רק לאחר קבלת רשות המשנית.
  8. יש לכתוב את התשובות בעט, בכתב יד ברור ונקי. נבחן הבוחר לכתוב טיוטה יעשה זאת בעמודו הימני של דפי מחברת הבחינה ויצוין בראש העמוד "טיוטה". אין לתלוש דפים מהמחברת.
  9. מחברות הבחינה שקיבל הנבחן תהיינה בפיקוחו ובאחריותו במשך כל הבחינה. בעת יציאה מן החדר יופקדו המחברות והשאלון בידי המשנית.
  10. בתום הבחינה יחזיר הנבחן את המחברות והשאלון ויקבל מידי המשנית את כרטיס הנבחן.
  11. המנהג בניגוד להוראות וליניהל סדרי בחינות ודיווח ציונים" צפוי להפסקת בחינתו ואף להעמדה לדין משמעתי.
  12. אין לכתוב מעבר לקו האדום משני צידי הדף.

בהצלחה.

תאריך הבחינה 13.2.04  
שם הקורס מונה 1  
שם המורה ס'ר'1  
החוג/המנחה מנהל ג'ס'2

52541

3



2

```

SparseMatrix * create( double **mat, int rows, int cols )
{
if ( rows < 1 || cols < 1 ) return null;
    SparseMatrix * newMat;
int rowNum;
    Row * newRow;
    Row * lastRow;

    if ( rows < 1 || cols < 1 ) return null;
    newMat = ( SparseMatrix * ) malloc( sizeof( SparseMatrix ) );
    if ( !newMat )
    {
        printf( "Error allocating memory!" );
        return NULL;
    }
    newMat->rows = rows;
    newMat->cols = cols;
    newMat->Row = NULL;
    for ( rowNum = 0; rowNum < rows; rowNum++ )
    {
        // create row
        // each row is a linked list of nodes
        // each node contains a value and a pointer to the next node
        // the first node is the head of the row
        // the last node is the tail of the row
        newRow = CreateRow( mat[rowNum], cols, rowNum );
        if ( newRow )
        {
            if ( !newMat->Row )
            {
                newMat->Row = newRow;
            }
            else
            {
                lastRow->next = newRow;
            }
            lastRow = newRow;
        }
    }
}
    
```



מגיש מוגש ע"י



מגיש מוגש ע"י



מגיש מוגש ע"י



מגיש מוגש ע"י

```

Row * CreateRow( double * rowArr, int length )
{
    return newMat;
}

Row * CreateRow( double * rowArr, int length )
{
    if ( lastRow ) lastRow -> next = NULL;
    return newMat;
}

```

הערה: למי בקושי להבחין בין המילים:

1000 Profit > Income Create Row 03/10/20 ~~11/10/20~~

ב'מזל      ג'ס'ח      דק'ט      זכ'יו      ח'יג      ט'פ'ו      י'ש'א

[illegible]

کتابت در ۱۳۰۸ هجری قمری در کتب خانۀ مدرسۀ عالیہ

Energy is conserved

5. בן ולסבורג, יונת, בת דודת דודת, יונת - 1917.

[illegible][illegible]

ایک روز ابراہیم علیہ السلام نے اپنے رب سے کہا کہ اے میرے رب! میں نے تجھ سے دعا کی ہے کہ تو میری قوم کو توحید پر لے آئے اور ان کو اپنے رب سے جدا کر دے۔

הכונן של הקובץ double de 23"מ"מ ומה שיש לי הוא 23"מ"מ. NULL

Row \* CreateRow( double \* row\_arr, int length, int rowNum ),  
{

Row \* newRow;

Entry \* newEntry;

Entry \* lastEntry;

Entry \* firstEntry = NULL;

int i;

for( i=0; i<length; i++ )

{

if( row\_arr[i] != 0 )

← אם זה לא 0 אז זה לא נכון

{

newEntry = (Entry \*) malloc( sizeof(Entry) );

if( !newEntry )

{

printf( "error allocating memory" );

exit -1;

}

if( !firstEntry )

newEntry → pos = i

newEntry → value = row\_arr[i]

firstEntry = newEntry;

else

lastEntry → next = newEntry;

lastEntry = newEntry;

}

ע. 32"מ"מ 23"מ"מ 23"מ"מ 23"מ"מ

}

if( !firstEntry )

←

הוא לא נכון ונכנס

return null;

else

{

newRow = (Row \*) malloc( sizeof(Row) );

if( !newRow ) { printf( "memory allocation error" ); exit -1; }

return newRow → index = rowNum;

newRow → Entry = firstEntry;

return newRow;

if( lastEntry )  
lastEntry → next  
= NULL;

3

3



הטבלה של הנתונים:

אנחנו מקבלים שורה של כל שורה בלתי-רציפה.

שורה כל שורה בלתי-רציפה וזו מופיעה ב-Creator

הפונקציה מיועדת לשימוש זה כל הנתונים באותו טבלה.

אנחנו כל הנתונים באותו טבלה וזו מופיעה ב-Creator.

start מופיע ב-Creator זה מופיע.

זה כל הנתונים בלתי-רציפה ב start בלתי-רציפה.

נחזור זה start בלתי-רציפה.

הנה, זה הנה בלתי-רציפה וזו מופיעה ב-Creator.

הנה מופיע ב entry בלתי-רציפה זה מופיע ב-Creator.

זה מופיע בלתי-רציפה זה מופיע ב-Creator.

במקום

הנתונים בלתי-רציפה  
ב-Creator

שורה ב'  
(בלתי-רציפה)

שורה ב'  
(בלתי-רציפה)

שורה ב'  
(בלתי-רציפה)

②

```
SparseMatrix * add( SparseMatrix *m1, SparseMatrix *m2 )
{
```

```
    SparseMatrix * newMat;
```

```
    Row * r1, * r2;
```

```
    Row * newRow;
```

```
    if ( (m1->rows != m2->rows) || (m1->cols != m2->cols) )
```

```
        return null;
```

כי לא יתאחד

```
    r1 = m1->head;
```

```
    r2 = m2->head;
```

}

הבדל בין שני המטריצות הוא רק בגודלן  
אם לא זהות הן לא יתאחדו

~~newMat~~

```
newMat = (SparseMatrix *) malloc( sizeof(SparseMatrix) );
```

```
if (!newMat)
```

```
{
```

```
    printf( "memory allocation error" );
```

```
    return NULL;
```

```
}
```

```
newMat->rows = m1->rows;
```

```
newMat->cols = m1->cols;
```

```
newMat->head = NULL;
```

```
while ( r1 || r2 )
```

```
{
```

```
    if ( r1 && r2 && (r1->index == r2->index) )
```

```
{
```

```
        newRow = addRow( r1, r2 );
```

```
    }
```

```
    r1 = r1->next; r2 = r2->next;
```

```
else
```

```
{
```

```
    if ( (r1 && r2) && (r1->index < r2->index) || (r1 && !r2) )
```

```
{
```

```
        newRow = addRow( r1, NULL );
```

```
    }
```

```
    r1 = r1->next;
```

```
else
```

```
{
```

```
        newRow = addRow( r2, NULL );
```

```
        r2 = r2->next;
```

```
    }
```

1

5

3



```

if (!newMat → head)
    newMat → head = newRow;
else
    lastRow → next = newRow;
lastRow = newRow;
}

if (lastRow)
    lastRow → next = NULL;

return newMat;
}

```

מיון  
שם  
row  
הוספת גרעין

הוספה:

אנו מחזיקים את הרשימה של החדשים.

שם 3 ופסול:

(א) הרשימה של החדשים היא רשימה פתוחה. יום כל יום נכנס ונכנס ונכנס

החדשים הם רשימה פתוחה של חדשים.

נקודת איתור של החדשים לחדשים חדשים וחדשים.

(ב) הרשימה של החדשים היא רשימה פתוחה של חדשים.

החדשים הם רשימה פתוחה של חדשים.

החדשים הם רשימה פתוחה של חדשים.

החדשים הם רשימה פתוחה של חדשים.

החדשים הם רשימה פתוחה של חדשים.

החדשים הם רשימה פתוחה של חדשים.

(ג) הרשימה של החדשים היא רשימה פתוחה של חדשים.

addRow סוגר בקוד קודי וכן גם הוריד וידו גם הוסר  
 וכן גם הוסר בקודי של NULL וכן גם הוסר וכן גם הוסר  
 וכן גם הוסר וכן גם הוסר.

Row \* addRow( Row \* r1, Row \* r2 )

{

Row \* newRow;

Entry \* newEntry;

Entry \* lastEntry;

~~Entry \* Entry \* e1, e2;~~

newRow = (Row \*) malloc ( sizeof(Row) );

newRow->index = r1->index;

newRow->Entry = NULL;

e1 = r1->head;

if ( r2 ) e1 = r2->head; else e1 = null;

while ( e1 || e2 )

{

~~if ( e1 || e2 ) {~~

newEntry = (Entry \*) malloc ( sizeof(Entry) );

if ( !newEntry )

{ printf ( "error allocating memory " );

exit -1;

}

~~newEntry->next = e1->next;~~

if ( e1 && e2 && ( e1->val == e2->val ) )

newEntry->val = e1->val;

newEntry->val = e1->val + e2->val;

} e1 = e1->next;

e2 = e2->next;

הוא  
 הוא  
 הוא

else

{

if (((e1 < e2) && (e1->next < e2->next)) || (e1 < e2))

{

newEntry->pos = e1->pos;

newEntry->value = e1->value;

}

e1 = e1->next;

}  
מקרה  
ש  
הערות  
הן  
הערות

else

{

newEntry->pos = e2->pos;

newEntry->value = e2->value;

e2 = e2->next;

}  
מקרה  
ש  
הערות  
הן  
הערות

}

}

if (!newRow->entry)

newRow->entry = newEntry

else

lastEntry = newEntry;

lastEntry = newEntry;

}

}  
מקרה  
ש  
הערות  
הן  
הערות

15/15

if (lastEntry)

lastEntry->next = NULL;

return newRow;

}

הבעיה היא להוסיף עמודה בלוח. כלומר, להוסיף עמודה (עמודה 3 בלוח).

אנחנו רוצים להוסיף עמודה בלוח. כלומר, להוסיף עמודה (עמודה 3 בלוח).

אנחנו רוצים להוסיף עמודה בלוח. כלומר, להוסיף עמודה (עמודה 3 בלוח).

אנחנו רוצים להוסיף עמודה בלוח. כלומר, להוסיף עמודה (עמודה 3 בלוח).

אנחנו רוצים להוסיף עמודה בלוח. כלומר, להוסיף עמודה (עמודה 3 בלוח).

③ void reversebits( char \* ch )  
{

char temp = 0; ←

(char bit; int i)

/\* ... \*/

/\* ... \*/

/\* ... \*/

~~bit = (ch & 1);~~

~~temp = (temp << 1) | bit;~~

/\* ... \*/

for ( i=0; i< sizeof(char)\*8; i++)

{ temp = temp << 1;

bit = (ch & 1);

temp = (temp | bit);

~~temp = temp << 1;~~

ch = ch >> 1;

}

/\* ... \*/

\*ch = temp;

}

✓

20/20

```

#include <stdio.h>
#include <string.h>
#include <malloc.h>

```

```

if ( ! newfilemane )
{
    printf ( "error allocating memory" );
    exit -1;
}

```

100 100 100  
 100 100 100  
 100 100 100  
 100 100 100

① ~~#define NUM\_CHARS 99~~  
 void write\_bits\_string( ~~char~~ char, FILE\* );  
 int main( int argc, char\*\* argv )  
 {

if ( argc != 2 )  
 {  
 printf( " 'הנני צריך שני' " );  
 exit -1;  
 }

FILE\* fip, \*fout; char c;  
 char\* newfilename;

newfilename = malloc( strlen( argv[1] ) + 3 );

strcpy( newfilename, argv[1] );  
 strcat( newfilename, ".r" );

fin = fopen( argv[1], "r" );

~~while ( fgetc( fin ) != EOF )~~  
 fout = fopen( newfilename, "w" );  
 free( newfilename );

while ( ( c = fgetc( fin ) ) != EOF )  
 {  
 write\_bits( c );  
 write\_bits\_string( c, fout );  
 }

fclose( fin );  
 fclose( fout );

}

לפי זה  
 המערכת  
 תפיק  
 (2)

הוא לא  
 יכול  
 להכיל  
 את  
 הסוף

הוא לא  
 יכול  
 להכיל  
 את  
 הסוף

הוא לא  
 יכול  
 להכיל  
 את  
 הסוף

הוא לא  
 יכול  
 להכיל  
 את  
 הסוף



הערה:

וידוע וזו בולק זאב מסעו הערצום דער חילק: דער פאך פאר און  
דער פאך פאר און דער פאך פאר און.

אוי צינג אקקונג מיטלע תוספת זינגן אים בקולאג האט.  
אסא בקולאג האט יעדא אים געקויפט אים אומאויס אים "ר".  
אוי געמאלט אים אסאדא לויט אים בקולאג. יאמא שטינג  
אסאדא יאמא אים אים (אויטע זינג אים אים אים אים  
אקקונג (אויטע)

Ques:- Write a program to find the sum of all even numbers from 1 to 10.

שקדלר מ' אהרנה' שקדלר.  
שקדלר מ' אהרנה' שקדלר.  
שקדלר מ' אהרנה' שקדלר.

```
void write_bits_string( char c , File & f )
```

```
{
```

```
    int i;    int bit;
```

```
    for ( i=0; i= sizeof(char)*8; i++ )    ← נוסף 8 ביט
```

```
{
```

```
    bit = ( c & 128 ),    ← נוסף 128
```

```
    if ( bit )
```

```
        fputc( '1', f );
```

```
    else
```

```
        fputc( '0', f );
```

```
    bit = bit << 1;    ← bit נוסף 1
```

```
}
```

```
}
```

↓  
c = c << 1  
3 ביט נוספים  
(-1)

✓  
המקרה של 0, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, 262144, 524288, 1048576, 2097152, 4194304, 8388608, 16777216, 33554432, 67108864, 134217728, 268435456, 536870912, 1073741824, 2147483648, 4294967296, 8589934592, 17179869184, 34359738368, 68719476736, 137438953472, 274877906944, 549755813888, 1099511627776, 2199023255552, 4398046511104, 8796093022208, 17592186044416, 35184372088832, 70368744177664, 140737488355328, 281474976710656, 562949953421312, 1125899906842624, 2251799813685248, 4503599627370496, 9007199254740992, 18014398509481984, 36028797018963968, 72057594037927936, 144115188075855872, 288230376151711744, 576460752303423488, 1152921504606846976, 2305843009213693952, 4611686018427387904, 9223372036854775808, 18446744073709551616, 36893488147419103232, 73786976294838206464, 147573952589676412928, 295147905179352825856, 590295810358705651712, 1180591620717411303424, 2361183241434822606848, 4722366482869645213696, 9444732965739290427392, 18889465931478580854784, 37778931862957161709568, 75557863725914323419136, 151115727451828646838272, 302231454903657293676544, 604462909807314587353088, 1208925819614629174706176, 2417851639229258349412352, 4835703278458516698824704, 9671406556917033397649408, 19342813113834066795298816, 38685626227668133590597632, 77371252455336267181195264, 154742504910672534362390528, 309485009821345068724781056, 618970019642690137449562112, 1237940039285380274899124224, 2475880078570760549798248448, 4951760157141521099596496896, 9903520314283042199192993792, 19807040628566084398385987584, 39614081257132168796771975168, 79228162514264337593543950336, 158456325028528675187087900672, 316912650057057350374175801344, 633825300114114700748351602688, 1267650600228229401496703205376, 2535301200456458802993406410752, 5070602400912917605986812821504, 10141204801825835211973625643008, 20282409603651670423947251286016, 40564819207303340847894502572032, 81129638414606681695789005144064, 162259276829213363391578010288128, 324518553658426726783156020576256, 649037107316853453566312041152512, 1298074214633706907132624082305024, 2596148429267413814265248164610048, 5192296858534827628530496329220096, 10384593717069655257060992658440192, 20769187434139310514121985316880384, 41538374868278621028243970633760768, 83076749736557242056487941267521536, 166153499473114484112975882535043072, 332306998946228968225951765070086144, 664613997892457936451903530140172288, 1329227995784915872903807060280344576, 2658455991569831745807614120560689152, 5316911983139663491615228241121378304, 10633823966279326983230456482242756608, 21267647932558653966460912964485513216, 42535295865117307932921825928971026432, 85070591730234615865843651857942052864, 170141183460469231731687303715884105728, 340282366920938463463374607431768211456, 680564733841876926926749214863536422912, 1361129467683753853853498429727072845824, 2722258935367507707706996859454145691648, 5444517870735015415413993718908291383296, 10889035741470030830827987437816582766592, 21778071482940061661655974875633165533184, 43556142965880123323311949751266331066368, 87112285931760246646623899502532662132736, 174224571863520493293247799005065324265472, 348449143727040986586495598010130648530944, 696898287454081973172991196020261297061888, 1393796574908163946345982392040522594123776, 2787593149816327892691964784081045188247552, 5575186299632655785383929568162090376495104, 11150372599265311570767859136324180752990208, 22300745198530623141535718272648361505980416, 44601490397061246283071436545296723011960832, 89202980794122492566142873090593446023921664, 178405961588244985132285746181186892047843328, 356811923176489970264571492362373784095686656, 713623846352979940529142984724747568191373312, 1427247692705959881058285969449495136382746624, 2854495385411919762116571938898990272765493248, 5708990770823839524233143877797980545530986496, 11417981541647679048466287755595961091061972992, 22835963083295358096932575511191922182123945984, 45671926166590716193865151022383844364247891968, 91343852333181432387730302044767688728495783936, 182687704666362864775460604089535377456991567872, 365375409332725729550921208179070754913983135744, 730750818665451459101842416358141509827966271488, 1461501637330902918203684832716283019655932542976, 2923003274661805836407369665432566039311865085952, 5846006549323611672814739330865132078623730171904, 11692013098647223345629478661730264157247460343808, 23384026197294446691258957323460528314494920687616, 46768052394588893382517914646921056628989841375232, 93536104789177786765035829293842113257979682750464, 187072209578355573530071658587684226515959365500928, 374144419156711147060143317175368453031918731001856, 748288838313422294120286634350736906063837462003712, 1496577676626844588240573268701473812127674924007424, 2993155353253689176481146537402947624255349848014848, 5986310706507378352962293074805895248510699696029696, 11972621413014756705924586149611790497021399392059392, 23945242826029513411849172299223580994042798784118784, 47890485652059026823698344598447161988085597568237568, 95780971304118053647396689196894323976171195136475136, 191561942608236107294793378393788647952342390272950272, 383123885216472214589586756787577295904684780545900544, 766247770432944429179173513575154591809369561091801088, 1532495540865888858358347027150309183618739122183602176, 3064991081731777716716694054300618367237478244367204352, 6129982163463555433433388108601236734474956488734408704, 12259964326927110866866776217202473468949912977468817408, 24519928653854221733733552434404946937899825954937634816, 49039857307708443467467104868809893875799651909875269632, 98079714615416886934934209737619787751599303819750539264, 196159429230833773869868419475239575503198607639501078528, 392318858461667547739736838950479151006397215279002157056, 784637716923335095479473677900958302012794430558004314112, 1569275433846670190958947355801916604025588861116008628224, 3138550867693340381917894711603833208051177722232017256448, 6277101735386680763835789423207666416102355444464034512896, 12554203470773361527671578846415332832204710888928069025792, 25108406941546723055343157692830665664409421777856138051584, 50216813883093446110686315385661331328818843555712276103168, 100433627766186892221372630771322662657637687111424552206336, 200867255532373784442745261542645325315275374222849104412672, 401734511064747568885490523085290650630550748445698208825344, 803469022129495137770981046170581301261101496891396417650688, 1606938044258990275541962092341162602522202993782792835301376, 3213876088517980551083924184682325205044405987565585670602752, 6427752177035961102167848369364650410088811975131171341205504, 12855504354071922204335696738729300820177623950262342682411008, 25711008708143844408671393477458601640355247900524685364822016, 51422017416287688817342786954917203280710495801049370729644032, 102844034832575377634685573909834406561420991602098741459288064, 205688069665150755269371147819668813122841983204197482918576128, 411376139330301510538742295639337626245683966408394965837152256, 822752278660603021077484591278675252491367932816789931674304512, 1645504557321206042154969182557350504982735865633579863348609024, 3291009114642412084309938365114701009965471731267159726697218048, 6582018229284824168619876730229402019930943462534319453394436096, 13164036458569648337239753460458804039861886925068638906788872192, 26328072917139296674479506920917608079723773850137277813577744384, 52656145834278593348959013841835216159447547700274555627155488768, 105312291668557186697918027683670432318895095400549111254310977536, 210624583337114373395836055367340864637790190801098222508621955072, 421249166674228746791672110734681729275580381602196445017243910144, 842498333348457493583344221469363458551160763204392890034487820288, 1684996666696914987166688442938726917102321526408785780068975640576, 3369993333393829974333376885877453834204643052817571560137951281152, 6739986666787659948666753771754907668409286105635143120275902562304, 13479973333575319897333507543509815336818572211270286240551805124608, 26959946667150639794667015087019630673637144422540572481103610249216, 53919893334301279589334030174039261347274288845081144962207220498432, 107839786668602559178668060348078522694548577690162289924414440996864, 215679573337205118357336120696157045389097155380324579848828881993728, 431359146674410236714672241392314090778194310760649159697657763987456, 862718293348820473429344482784628181556388621521298319395315527974912, 1725436586697640946858688965569256363112777243042596638790631055949824, 3450873173395281893717377931138512726225554486085193277581262111899648, 6901746346790563787434755862277025452451108972170386555162524223799296, 13803492693581127574869511724554050904902217944340773110325048447598592, 27606985387162255149739023449108101809804435888681546220650096895197184, 55213970774324510299478046898216203619608871777363092441300193790394368, 110427941548649020598956093796432407239217743554726184882600387580788736, 220855883097298041197912187592864814478435487109452369765200775161577472, 441711766194596082395824375185729628956870974218904739530401550323154944, 883423532389192164791648750371459257913741948437809479060803100646309888, 1766847064778384329583297500742918515827483896875618958121606201292619776, 3533694129556768659166595001485837031654967793751237916243212402585239552, 7067388259113537318333190002971674063309935587502475832486424805170479104, 14134776518227074636666380005943348126619871175004951664972849610340958208, 28269553036454149273332760011886696253239742350009903329945699220681916416, 56539106072908298546665520023773392506479484700019806659891398441363832832, 113078212145816597093331040047546785012958969400039613319782796882727665664, 226156424291633194186662080095093570025917938800079226639565593765455331328, 452312848583266388373324160190187140051835877600158453279131187530910662656, 904625697166532776746648320380374280103671755200316906558262375061821325312, 1809251394333065553493296640760748560207343510400633813116524750123642650624, 3618502788666131106986593281521497120414687020801267626233049500247285301248, 7237005577332262213973186563042994240829374041602535252466099000494570602496, 14474011154664524427946373126085988481658748083205070504932198000989141204992, 28948022309329048855892746252171976963317496166410141009864396001978282409984, 57896044618658097711785492504343953926634992332820282019728792003956564819968, 115792089237316195423570985008687907853269984665640564039457584007913129639936, 231584178474632390847141970017375815706539969331281128078915168015826259279872, 463168356949264781694283940034751631413079938662562256157830336031652518559744, 926336713898529563388567880069503262826159877325124512315660672063305037119488, 1852673427797059126777135760139006525652319754650249024631321344126610074238976, 3705346855594118253554271520278013051304639509300498049262642688253220148477952, 7410693711188236507108543040556026102609279018600996098525285376506440296955904, 14821387422376473014217086081112052205218558037201992197050570753012880593911808, 29642774844752946028434172162224104410437116074403984394101141506025761187823616, 59285549689505892056868344324448208820874232148807968788202283012051522375647232, 118571099379011784113736688648896417641748464297615937576404566024103044751294464, 237142198758023568227473377297792835283496928595231875152809132048206089502588928, 474284397516047136454946754595585670566993857190463750305618264096412179005177856, 948568795032094272909893509191171341133987714380927500611236528192824358010355712, 189713759006418854581978701838234268226797542876185500122247