שאלה 1 (20 נקודות)

בדומה לסדרת פיבונצ'י, סדרת אבו-נצ'י הינה סדרת מספרים בה כל איבר הינו סכום שני האיברים שלפניו. ההבדל הוא ששני האיברים הראשונים בסדרה אינם בהכרח 1 אולם הם בהכרח חד-ספרתיים. למשל, הסדרה הבאה היא סדרת אבו-נצ'י חוקית: 1, 9, 10, 19, 29, 48, …
ניתן להפוך כל מספר למחרוזת ולשרשר את אוסף המחרוזות לקבלת מחרוזת אחת. לדוגמא, עבור הסדרה הנ"ל נקבל את המחרוזת "1910192948".
יש לכתוב פונקציה המקבלת מחרוזת המורכבת מספרות ובודקת האם היא מכילה מספרים המקיימים את תנאי סדרת אבו-נצ'י. למשל על הפונצקיה להשיב אמת בהינתן המחרוזת "1910192948" ושקר בהנתן המחרוזת "15712195069".
הערות:

- ניתן להניח כי המחרוזת מכילה ספרות בלבד.
- מחרוזת בת שתי ספרות בלבד מקיימת את התנאי (באופן ריק).
- ניתן להניח כי במחרוזת יש לפחות שתי ספרות.
- ניתן להניח כי אורך המחרוזת המקסימלית הוא N המוגדר ב #define.
- אין להשתמש בפונקציות ספריה.

א. ממשו את הפונקציה num2str, אשר מקבלת מספר שלם אי-שלילי num ומחזירה מחרוזת str המייצגת את אותו מספר בפורמט עשרוני (ניתן להניח כי הוקצה מספיק מקום עבור אותיות המחרוזת).

```
void num2str(unsigned int num, char *str) {

  int dignum = 1, i;
  unsigned int tempnum = num;

  while (tempnum>9) {
    tempnum /= 10;
    dignum++;
  }

  str[dignum] = 0;
  while (dignum>0) {
    dignum--;
    str[dignum] = num%10 + '0';
    num /= 10;
  }
}
```

ב. ממשו את הפונקציה beginswith אשר משווה בין מחרוזת A למחרוזת B. הפונקציה תחזיר 1 אם המחרוזת B מתחילה במחרוזת A, ו-0 אחרת. לדוגמא: עבור "A="abc ו "B="abcdef הפונקציה תחזיר 1, ועבור "B="acbdef הפונקציה תחזיר 0. אם A ריקה, הפונקציה תחזיר 1.

```
unsigned int beginswith (char *A, char *B) {

  while (*A && *B) {
    if (*A != *B)
       break;
    A++; B++;
  }

  if (*A == 0)
    return 1;

  return 0;
}
```

ג. ממשו את הפונקציה IsAbuNacci, אשר מחזירה 1 אם מחרוזת הקלט str מקיימת את תנאי סדרת אבו-נצ'י, ו-0 אחרת. מותר (ואף מומלץ) להשתמש בפונקציות שהוגדרו בסעיפים א' ו-ב', גם אם לא פתרתם אותם.

```
unsigned int IsAbuNacci(char *str) {

  int num1, num2, pos=0;
  char nextnum[N];

  num1 = str[pos++] - '0';
  num2 = str[pos++] - '0';

  while(str[pos]) {
    num2str(num1+num2, nextnum);
    if (!beginswith(nextnum, str+pos))
      return 0;
    pos += strlen(nextnum);
    num2 = num1+num2;
    num1 = num2-num1;
  }
  return 1;
}
```

```
int strlen(char *str)
{
  int len = 0;
  while (str[len])
    len++;
  return len;
}
```