

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ  
UNIVERSITY OF WEST ATTICA

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ  
ΥΠΟΛΟΓΙΣΤΩΝ

## ΕΡΓΑΣΙΑ 6 ΠΙΝΑΚΕΣ-ΔΕΙΚΤΕΣ-ΑΡΧΕΙΑ

### ΣΤΟΙΧΕΙΑ ΦΟΙΤΗΤΗ:

---

**ΟΝΟΜΑΤΕΠΩΝΥΜΟ:** ΑΘΑΝΑΣΙΟΥ ΒΑΣΙΛΕΙΟΣ ΕΥΑΓΓΕΛΟΣ

**ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ:** 19390005

**ΚΑΤΑΣΤΑΣΗ ΦΟΙΤΗΤΗ:** ΠΡΟΠΤΥΧΙΑΚΟ

**ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ:** ΠΑΔΑ

**ΤΜΗΜΑ ΕΡΓΑΣΤΗΡΙΟΥ:** Μ2

**ΚΑΘΗΓΗΤΗΣ ΕΡΓΑΣΤΗΡΙΟΥ:** ΓΕΩΡΓΙΟΣ ΜΕΛΕΤΙΟΥ

**ΗΜΕΡΟΜΗΝΙΑ ΟΛΟΚΛΗΡΩΣΗΣ ΤΗΣ ΕΡΓΑΣΙΑΣ:** 31/1/2022

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## ΠΗΓΑΙΟΙ ΚΩΔΙΚΕΣ / ΤΕΚΜΗΡΙΩΣΕΙΣ

### ΘΕΜΑ 2

#### ΕΠΙΣΗΜΑΝΣΗ «Combinations.c»

Το «Πρόγραμμα “Combinations.c”» (Πηγαίος Κώδικας) και η «Τεκμηρίωση “Combinations.c”» (Ζητούμενο, Δομή, Συναρτήσεις, Μεταβλητές, Διάσχιση, Παραδείγματα, Παρατηρήσεις) απαντούν στο ζητούμενο του ερωτήματος «Θέμα 2».

#### ΠΡΟΓΡΑΜΜΑ «Combinations.c»

```
#include <stdio.h>

#include <stdlib.h>

/* Δήλωση συναρτήσεων */

void Title (); // Ο τίτλος του προγράμματος

int Read_N_Numbers (int, int); // Εισαγωγή του πλήθους "N" των αριθμών, όπου
το "N" ανήκει στο διάστημα (6, 49]

void Print_N (int N); // Εκτύπωση του πλήθους των αριθμών στο διάστημα (6, 49]

void Read_Matrix (int *, int); // Εισαγωγή των "N" αριθμών στο διάστημα [1,
49] και εκχώρηση σε δυναμικά δεσμευμένο πίνακα

int Search (int *, int, int, int, int); // Αναζήτηση στον δυναμικά δεσμευμένο
πίνακα με τους αριθμούς για πιθανόν εντοπισμό ίδιων καταχωρημένων αριθμών

int *Read_Even_Lims (); // Εισαγωγή του διαστήματος [0, X1]U[X1, X2]U[X2, 6]
για το πλήθος των άρτιων αριθμών κάθε συνδυασμού

int *Read_Sum_Lims (); // Εισαγωγή του διαστήματος [21, Y1]U[Y1, Y2]U[Y2,
279] για το άθροισμα των αριθμών κάθε συνδυασμού

void Check_Memory (int *); // Έλεγχος στη μνήμη για τις δυναμικές δεσμεύσεις
πινάκων

void Sort (int *, int); // Ταξινόμηση του πίνακα με τους αριθμούς κατά αύξουσα
σειρά

void Print_Matrix (int *, int); // Εκτύπωση του πίνακα με τους "N" αριθμούς
στο διάστημα [1, 49]

void Print_Even_Lims (int *); // Εκτύπωση του διαστήματος [0, X1]U[X1,
X2]U[X2, 6] για το πλήθος των άρτιων αριθμών κάθε συνδυασμού

void Print_Sum_Lims (int *); // Εκτύπωση του διαστήματος [21, Y1]U[Y1,
Y2]U[Y2, 279] για το άθροισμα των αριθμών κάθε συνδυασμού

int *Combos (int *, int, int *, int *); // Δημιουργία των συνδυασμών

int Search_Evens (int *, int *, int); // Αναζήτηση για το πλήθος των άρτιων
αριθμών κάθε συνδυασμού

int Search_Sum (int *, int *, int); // Αναζήτηση για το άθροισμα των αριθμών
κάθε συνδυασμού

void Print_Combos (int *); // Εκτύπωση των συνδυασμών
```

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
void Frequency (int *, int *); // Υπολογισμός της συχνότητας εμφάνισης κάθε
αριθμού στους εκτυπωμένους συνδυασμούς

void Print_Freq (int *); // Εκτύπωση της συχνότητας εμφάνισης κάθε αριθμού
στος εκτυπωμένους συνδυασμούς

void Print_num_Combos (int *); // Εκτύπωση του πλήθους των συνδυασμών που
πληρούν συγκεκριμένους όρους

void Free_Memory (int *); // Αποδέσμευση μνήμης


int main (int argc, char **argv) /* main (int argc, char **argv) */
{

    system ("chcp 1253");

    int *Mat; // Δήλωση μεταβλητών

    int N;

    int *Lim_s, *Lim_e;

    int *c;


    N = Read_N_Numbers (6, 49); // Κλήση της συνάρτησης "Read_N_Numbers (6,
49)"

    Mat = (int *) malloc (N * sizeof (int)); // Δυναμική δέσμευση μνήμης
για την δημιουργία ενός μονοδιάστατου πίνακα "Mat" με τους εισαχθέντες αριθμούς
στο διάστημα [1, 49]

    Check_Memory (Mat); // Κλήση της συνάρτησης "Check_Memory (Mat)"

    Read_Matrix (Mat, N); // Κλήση της συνάρτησης "Read_Matrix (Mat, N)"

    Sort (Mat, N); // Κλήση της συνάρτησης "Sort (Mat, N)"

    Lim_e = Read_Even_Lims (); // Κλήση της συνάρτησης "Read_Even_Lims ()"

    Lim_s = Read_Sum_Lims (); // Κλήση της συνάρτησης "Read_Sum_Lims ()"

    Print_Matrix (Mat, N); // Κλήση της συνάρτησης "Print_Matrix (Mat, N)"

    Print_N (N); // Κλήση της συνάρτησης "Print_N (N)"

    Print_Even_Lims (Lim_e); // Κλήση της συνάρτησης "Read_Even_Lims
(Lim_e)"

    Print_Sum_Lims (Lim_s); // Κλήση της συνάρτησης "Read_Sum_Lims
(Lim_s)"

    c = Combos (Mat, N, Lim_e, Lim_s); // Κλήση της συνάρτησης "Combos (Mat,
N, Lim_e, Lim_s)"

    return 0;

}
```

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
void Title () /* Title () */
{
    printf ("=====
\n\n");
    printf ("ΣΥΝΔΥΑΣΜΟΙ\n\n"); // Ο τίτλος του προγράμματος
    printf ("=====
\n\n");
}

int Read_N_Numbers (int L1, int L2) /* Read_N_Numbers (6, 49) */
{
    int x; // Δήλωση μεταβλητών
    do /* Βρόχος */
    {
        system ("cls");
        Title (); // Κλήση της συνάρτησης "Title ()"
        printf ("Εισάγετε το πλήθος των αριθμών 'N' στο διάστημα (6, 49]
: ");
        scanf ("%d", &x); // Εισαγωγή του πλήθους "N" των αριθμών, όπου
το "N" ανήκει στο διάστημα (6, 49]
        printf ("\n-----
-----\n\n");
    }
    while (x <= L1 || x > L2);

    return x; // Επιστροφή του πλήθους "N" των αριθμών, όπου το "N" ανήκει
στο διάστημα (6, 49]
}

void Print_N (int N) /* Print_N (N) */
{
    printf ("N : %3d\n\n", N); // Εκτύπωση του πλήθους των αριθμών στο
διάστημα (6, 49]
}

void Read_Matrix (int *M, int N) /* Read_Matrix (Mat, N) */
{
    int i; // Δήλωση μεταβλητών
    for (i = 0; i < N; i++) /* 1ος Βρόχος */
    {
```

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
do /* 2ος Βρόχος */
{
    printf ("Εισάγετε αριθμό : ");

    scanf ("%d", R + i); // Εισαγωγή των "N" αριθμών στο
    διάστημα [1, 49] και εκχώρηση στον δυναμικά δεσμευμένο πίνακα "Mat"

}

while (Search (M, i, *(M + i), 1, 49) != -1); // Κλήση της
συνάρτησης "Search (Mat, 0..N, *(Mat + 0..N), 1, 49)"

}

}

int Search (int *M, int N, int Num, int O1, int O2) /* Search (Mat, 0..N,
*(Mat + 0..N), 1, 49) */
{
    int j = 0; // Δήλωση μεταβλητών

    if (N == 0) /* (~) Θέση πίνακα *Mat */
    {
        if (Num >= O1 && Num <= O2) /* (!) Ο αριθμός ανήκει στο διάστημα
[1, 49] */
            return -1; // Επιστροφή τιμής επιβεβαίωσης (αποδεκτός
αριθμός)

        return j; // Επιστροφή τιμής σφάλματος (απορριπτέος αριθμός)
    }

    else /* (~) Θέσεις πίνακα από *(Mat + 1) και μετά */
    {
        while (Num >= O1 && Num <= O2 && j < N && *(M + j) != Num) /*
        Βρόχος αναζήτησης για τον εντοπισμό πιθανών ισοτήτων στα περιεχόμενα του πίνακα
        και περιεχομένων που δεν ανήκουν στο διάστημα [1, 49]*/

            j++; // Άυξηση της βοηθητικής μεταβλητής για την παραβίαση
            των περιορισμών καταχώρησης ενός αριθμού στον πίνακα "Mat"

            if (j == N) /* (+) Η βοηθητική μεταβλητή είναι ίση με την θέση
            του πίνακα που έχει φτάσει η αναζήτηση */
                return -1; // Επιστροφή τιμής επιβεβαίωσης (αποδεκτός
                αριθμός)

            return j; // Επιστροφή τιμής σφάλματος (απορριπτέος αριθμός)
        }
    }

}

int *Read_Even_Lims () /* Read_Even_Lims () */
{
    int *Evens; // Δήλωση μεταβλητών
```

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
Evens = (int *) malloc (2 * sizeof (int)); // Δυναμική δέσμευση μνήμης
για την δημιουργία ενός μονοδιάστατου πίνακα "Evens" με τους δύο εισαχθέντες
αριθμούς στο διάστημα [0, X1]U[X1, X2]U[X2, 6] για το πλήθος των άρτιων αριθμών
κάθε συνδυασμού
```

```
Check_Memory (Evens); // Κλήση της συνάρτησης "Check_Memory (Evens)"

do /* Βρόχος */
{
    system ("cls");

    printf ("Εισάγετε τα όρια των άρτιων αριθμών του κάθε συνδυασμού
στο διάστημα [0, X1]U[X1, X2]U[X2, 6]\n\n");

    printf ("X1 : ");

    scanf ("%d", Evens); // Εισαγωγή του πρώτου ορίου "X1" του
διαστήματος [0, X1]U[X1, X2] για το πλήθος των άρτιων αριθμών κάθε συνδυασμού

    printf ("X2 : ");

    scanf ("%d", Evens + 1); // Εισαγωγή του δεύτερου ορίου "X2" του
διαστήματος [X1, X2]U[X2, 6] για το πλήθος των άρτιων αριθμών κάθε συνδυασμού

    printf ("\n");

}

while (*Evens < 0 || *Evens > *(Evens + 1) || *(Evens + 1) > 6);

return Evens; // Επιστροφή του πίνακα "Evens" με τα όρια "X1" και "X2"
}
```

```
int *Read_Sum_Lims () /* Read_Sum_Lims () */
{
    int *Sum; // Δήλωση μεταβλητών

    Sum = (int *) malloc (2 * sizeof (int)); // Δυναμική δέσμευση μνήμης
για 1Δ πίνακα "Sum" με τους δύο εισαχθέντες αριθμούς στο διάστημα [21, Y1]U[Y1,
Y2]U[Y2, 279] για το άθροισμα αριθμών κάθε συνδυασμού

    Check_Memory (Sum); // Κλήση της συνάρτησης "Check_Memory (Sum)"

    do /* Βρόχος */
    {
        system ("cls");

        printf ("Εισάγετε τα όρια του αθροίσματος των αριθμών κάθε
συνδυασμού στο διάστημα [21, Y1]U[Y1, Y2]U[Y2, 279]\n\n");

        printf ("Y1 : ");

        scanf ("%d", Sum); // Εισαγωγή του πρώτου ορίου "Y1" του
διαστήματος [21, Y1]U[Y1, Y2] για το άθροισμα των αριθμών κάθε συνδυασμού

        printf ("Y2 : ");

        scanf ("%d", Sum + 1); // Εισαγωγή του δεύτερου ορίου "Y2" του
διαστήματος [Y1, Y2]U[Y2, 279] για το άθροισμα των αριθμών κάθε συνδυασμού
```

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
        printf ("\n");
    }

    while (*Sum < 21 || *Sum > *(Sum + 1) || *(Sum + 1) > 279);

    return Sum; // Επιστροφή του πίνακα "Sum" με τα όρια "X1" και "X2"
}

void Check_Memory (int *Array) /* Check_Memory (Array) */
{
    if (Array == NULL) /* (~) Ο δείκτης "Array" δείχνει στον δείκτη "NULL"
*/
    {
        system ("cls");
        printf ("\nΠρόβλημα στη δέσμευση μνήμης\n\n");
        exit (1); // Τερματισμός του προγράμματος με επιστρεφόμενη τιμή
"1"
    }
}

void Sort (int *M, int N) /* Sort (Mat, N) */
{
    int i, tmp, j; // Δήλωση μεταβλητών
    for (i = 0; i < N - 1; i++) /* 1ος Βρόχος */
        for (j = i + 1; j < N; j++) /* 2ος Βρόχος */
            if (*(M + i) > *(M + j)) /* (~) *M == 21 > *(M + 1) == 3
*/
            {
                tmp = *(M + i); // tmp == *M ==> tmp == 21
                *(M + i) = *(M + j); // *M == *(M + 1) ==> *M == 3
                *(M + j) = tmp; // *(M + 1) == tmp ==> *(M + 1) ==
21
            }
}

void Print_Matrix (int *M, int N) /* Print_Matrix (Mat, N) */
{
    int i; // Δήλωση μεταβλητών
    system ("cls");
```

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
for (i = 0; i < N; i++) /* Βρόχος */
{
    printf ("Στοιχείο [%2d] Τιμή : %3d\n", i + 1, *(M + i)); //
    Εκτύπωση των στοιχείων του πίνακα "Mat" με τους "N" αριθμούς στο διάστημα [1,
    49] ταξινομημένους
}

printf ("\n");
}

void Print_Even_Lims (int *E) /* Print_Even_Lims (Evens) */
{
    int i; // Δήλωση μεταβλητών
    for (i = 0 ; i < 2; i++) /* Βρόχος */
    {
        if (i == 0) /* (~) Θέση *Evens, όπου ανήκει το X1 */
        {
            printf ("X1 : %3d\n", *(E + i)); // Εκτύπωση του πρώτου
            ορίου "X1" του διαστήματος [0, X1]U[X1, X2] για το πλήθος των άρτιων αριθμών
            κάθε συνδυασμού
        }
        else /* (~) Θέση *(Evens + 1) και μετά */
        {
            if (i == 1) /* (!) Θέση *(Evens + 1), όπου ανήκει το X2
            */
            {
                printf ("X2 : %3d\n\n", *(E + i)); // Εισαγωγή του
                δεύτερου ορίου "X2" του διαστήματος [X1, X2]U[X2, 6] για το πλήθος των άρτιων
                αριθμών κάθε συνδυασμού
            }
        }
    }
}

void Print_Sum_Lims (int *S) /* Print_Sum_Lims (Sum) */
{
    int j; // Δήλωση μεταβλητών
    for (j = 0 ; j < 2; j++) /* Βρόχος */
    {
        if (j == 0) /* (~) Θέση *Sum, όπου ανήκει το Y1 */
        {
```



# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
        printf ("Y1 : %3d\n", *(S + j)); // Εκτύπωση του πρώτου
ορίου "Y1" του διαστήματος [21, Y1]U[Y1, Y2] για το άθροισμα των αριθμών κάθε
συνδυασμού

    }

    else /* (~) Θέση *(Sum + 1) και μετά */

    {

        if (j == 1) /* (!) Θέση *(Sum + 1), όπου ανήκει το Y2 */

        {

            printf ("Y2 : %3d\n\n", *(S + j)); // Εκτύπωση του
δεύτερου ορίου "Y2" του διαστήματος [Y1, Y2]U[Y2, 279] για το άθροισμα των
αριθμών κάθε συνδυασμού

        }

    }

}

int *Combos (int *M, int N, int *E, int *S) /* Combos (Mat, N, Evens, Sum) */

{

    int i, j, k, l, m, n; // Δήλωση μεταβλητών

    int *Com, *Cnt, *Freq;

    int evens, sum;

    Com = (int *) malloc (6 * sizeof (int)); // Δυναμική δέσμευση μνήμης
για 1Δ πίνακα "Com" με τον κάθε συνδυασμό που δημιουργείται

    Check_Memory (Com); // Κλήση της συνάρτησης "Check_Memory (Com)"

    Cnt = (int *) calloc (5, sizeof (int)); // Δυναμική δέσμευση μνήμης για
1Δ πίνακα "Cnt" (με αρχική τιμή "0" σε κάθε θέση του πίνακα) με μετρητές για
τον υπολογισμό του πλήθους συνδυασμών που πληρούν συγκεκριμένους όρους

    Check_Memory (Cnt); // Κλήση της συνάρτησης "Check_Memory (Cnt)"

    Freq = (int *) calloc (49, sizeof (int)); // Δυναμική δέσμευση μνήμης
για 1Δ πίνακα "Freq" (με αρχική τιμή "0" σε κάθε θέση του πίνακα) με μετρητές
για τον υπολογισμό της συχνότητας εμφάνισης κάθε αριθμού στους τυπωμένους
συνδυασμούς

    Check_Memory (Freq); // Κλήση της συνάρτησης "Check_Memory (Freq)"

    for (i = 0; i < N - 5; i++) /* 1ος Βρόχος */

        for (j = i + 1; j < N - 4; j++) /* 2ος Βρόχος */

            for (k = j + 1; k < N - 3; k++) /* 3ος Βρόχος */

                for (l = k + 1; l < N - 2; l++) /* 4ος Βρόχος */

                    for (m = l + 1; m < N - 1; m++) /* 5ος Βρόχος

*/

                        for (n = m + 1; n < N; n++) /* 6ος

Βρόχος */
```

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
{  
    *(Com + 0) = *(M + i); // 1ος  
    αριθμός του συνδυασμού στη θέση *(Com + 0) του πίνακα "Com"  
    *(Com + 1) = *(M + j); // 2ος  
    αριθμός του συνδυασμού στη θέση *(Com + 1) του πίνακα "Com"  
    *(Com + 2) = *(M + k); // 3ος  
    αριθμός του συνδυασμού στη θέση *(Com + 2) του πίνακα "Com"  
    *(Com + 3) = *(M + l); // 4ος  
    αριθμός του συνδυασμού στη θέση *(Com + 3) του πίνακα "Com"  
    *(Com + 4) = *(M + m); // 5ος  
    αριθμός του συνδυασμού στη θέση *(Com + 4) του πίνακα "Com"  
    *(Com + 5) = *(M + n); // 6ος  
    αριθμός του συνδυασμού στη θέση *(Com + 5) του πίνακα "Com"  
    *Cnt = *Cnt + 1; // Αύξηση του  
    μετρητή για τον υπολογισμό του πλήθους των συνδυασμών "N" ανά 6  
    evens = Search_Evens (Com, E,  
6); // Κλήση της συνάρτησης "Search_Evens (Com, Evens, 6)"  
    sum = Search_Sum (Com, S, 6);  
    // Κλήση της συνάρτησης "Search_Sum (Com, Sum, 6)"  
    if (evens == 1) /* (~) Πλήθος  
    των άρτιων του συνδυασμού μέσα στα όρια [X1, X2] */  
    {  
        if (sum == 1) /* (!)  
        'θροισμα των αριθμών του συνδυασμού μέσα στα όρια [Y1, Y2] */  
        {  
            Print_Combos  
(Com); // Κλήση της συνάρτησης "Print_Combos (Com)"  
            *(Cnt + 3) =  
*(Cnt + 3) + 1; // Αύξηση του μετρητή για τον υπολογισμό του πλήθους των  
    συνδυασμών που τυπώθηκαν  
            Frequency (Freq,  
Com); // Κλήση της συνάρτησης "Frequency (Freq, Com)"  
        }  
        else /* (!) 'θροισμα των  
    αριθμών του συνδυασμού έξω από τα όρια [Y1, Y2] */  
        {  
            *(Cnt + 2) =  
*(Cnt + 2) + 1; // Αύξηση του μετρητή για τον υπολογισμό του πλήθους των  
    συνδυασμών που πληρούσαν τον πρώτο, αλλά δεν πληρούσαν τον δεύτερο όρο  
        }  
    }  
    else /* (~) Πλήθος των άρτιων  
    του συνδυασμού έξω από τα όρια [X1, X2] */  
    {
```

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
*(Cnt + 1) = *(Cnt + 1)
+ 1; // Αύξηση του μετρητή για τον υπολογισμό του πλήθους που δεν πληρούσαν
τον πρώτο όρο

    }

}

Print_num_Combos (Cnt); // Κλήση της
συνάρτησης "Print_num_Combos (N, Cnt)"

Print_Freq (Freq); // Κλήση της
συνάρτησης "Print_Freq (Freq)"

Free_Memory (M); // Κλήση της
συνάρτησης "Free_Memory (Mat)"

Free_Memory (E); // Κλήση της
συνάρτησης "Free_Memory (Evens)"

Free_Memory (S); // Κλήση της
συνάρτησης "Free_Memory (Sum)"

Free_Memory (Com); // Κλήση της
συνάρτησης "Free_Memory (Com)"

return Cnt; // Επιστροφή του πίνακα "Cnt" με τους μετρητές
}

int Search_Evens (int *C, int *E, int N) /* Search_Evens (Mat, Evens, N) */
{
    int j = 0; // Δήλωση μεταβλητών
    int evens = 0;
    while (j < N) /* Βρόχος */
    {
        if (*(C + j) % 2 == 0) /* (~) 'ρτιος αριθμός */
        {
            evens++; // Άυξηση της βοηθητικής μεταβλητής για το πλήθος
των άρτιων αριθμών

            j++; // Αύξηση της βοηθητικής μεταβλητής για την
προσπέλαση του συνδυασμού
        }
        else /* (~) Περιττός αριθμός */
        {
            j++; // Αύξηση της βοηθητικής μεταβλητής για την
προσπέλαση του συνδυασμού
        }
    }

    if (evens >= *E && evens <= *(E + 1)) /* (!) Το πλήθος των άρτιων αριθμών
ανήκει στα όρια [X1, X2] */
```

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
        return 1; // Επιστροφή τιμής επιβεβαίωσης
    return 0; // Επιστροφή τιμής απόρριψης
}

int Search_Sum (int *C, int *S, int N) /* Search_Sum (Mat, Sum, N) */
{
    int sum = 0; // Δήλωση μεταβλητών
    int j;
    for (j = 0; j < N; j++) /* Βρόχος */
        sum = sum + *(C + j); // Υπολογισμός του αθροίσματος των αριθμών
        του συνδυασμού

    if (sum >= *S && sum <= *(S + 1)) /* (~) Το άθροισμα των αριθμών του
    συνδυασμού ανήκει στα όρια [Y1, Y2] */
        return 1; // Επιστροφή τιμής επιβεβαίωσης
    return 0; // Επιστροφή τιμής απόρριψης
}

void Print_Combos (int *C) /* Print_Combos (Com) */
{
    printf ("%2d %2d %2d %2d %2d %2d\n", *(C + 0), *(C + 1), *(C + 2), *(C
+ 3), *(C + 4), *(C + 5)); // Εκτύπωση του έγκυρου συνδυασμού
}

void Frequency (int *F, int *C) /* Frequency (Freq, Com) */
{
    *(F + (*(C + 0) - 1)) = *(F + (*(C + 0) - 1)) + 1; // Αύξηση του μετρητή
για την συχνότητα εμφάνισης του "N" αριθμού της 1ης θέσης του συνδυασμού

    *(F + (*(C + 1) - 1)) = *(F + (*(C + 1) - 1)) + 1; // Αύξηση του μετρητή
για την συχνότητα εμφάνισης του "N" αριθμού της 2ης θέσης του συνδυασμού

    *(F + (*(C + 2) - 1)) = *(F + (*(C + 2) - 1)) + 1; // Αύξηση του μετρητή
για την συχνότητα εμφάνισης του "N" αριθμού της 3ης θέσης του συνδυασμού

    *(F + (*(C + 3) - 1)) = *(F + (*(C + 3) - 1)) + 1; // Αύξηση του μετρητή
για την συχνότητα εμφάνισης του "N" αριθμού της 4ης θέσης του συνδυασμού

    *(F + (*(C + 4) - 1)) = *(F + (*(C + 4) - 1)) + 1; // Αύξηση του μετρητή
για την συχνότητα εμφάνισης του "N" αριθμού της 5ης θέσης του συνδυασμού

    *(F + (*(C + 5) - 1)) = *(F + (*(C + 5) - 1)) + 1; // Αύξηση του μετρητή
για την συχνότητα εμφάνισης του "N" αριθμού της 6ης θέσης του συνδυασμού
}
```

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
void Print_Freq (int *F) /* Print_Freq (Freq) */
{
    int i; // Δήλωση μεταβλητών
    printf ("\n");
    for (i = 0; i < 49; i++) /* Βρόχος */
    {
        if (*(F + i) != 0) /* (~) Εκτύπωση στοιχείου του πίνακα "Freq"
που δεν περιέχει την τιμή "0" */
        {
            printf ("Συχνότητα του [%2d] : [%8d]\n", i + 1, *(F + i));
// Εκτύπωση της συχνότητας εμφάνισης του κάθε "N" αριθμού που επιλέχθηκε για
την δημιουργία συνδυασμών
        }
    }

    Free_Memory (F); // Κλήση της συνάρτησης "Free_Memory (Freq)"
}

void Print_num_Combos (int *Cnt) /* Print_num_Combos (Cnt) */
{
    printf ("Το πλήθος των συνδυασμών N ανά 6 είναι
: [%8d]\n", N, *Cnt); // Εκτύπωση του πλήθους των συνδυασμών N ανά 6 είναι

    printf ("Το πλήθος των συνδυασμών που δεν πληρούσαν τον πρώτο όρο
: [%8d]\n", *(Cnt + 1)); // Εκτύπωση του πλήθους των συνδυασμών που δεν
πληρούσαν τον πρώτο όρο

    printf ("Το πλήθος των συνδυασμών που πληρούσαν τον πρώτο, αλλά δεν
πληρούσαν τον δεύτερο όρο : [%8d]\n", *(Cnt + 2)); // Εκτύπωση του πλήθους των
συνδυασμών που πληρούσαν τον πρώτο, αλλά δεν πληρούσαν τον δεύτερο όρο

    printf ("Το πλήθος των συνδυασμών που τυπώθηκαν
: [%8d]\n", *(Cnt + 3)); // Εκτύπωση του πλήθους των συνδυασμών που τυπώθηκαν

    Free_Memory (Cnt); // Κλήση της συνάρτησης "Free_Memory (Cnt)"
}

void Free_Memory (int *Array) /* Free_Memory (Array) */
{
    free (Array); // Αποδέσμευση μνήμης που χρησιμοποιήθηκε για την
δημιουργία δυναμικού πίνακα
}
```

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## ΤΕΚΜΗΡΙΩΣΗ «Combinations.c»

### **ΖΗΤΟΥΜΕΝΟ**

Το πρόγραμμα «Combinations.c» επιτυγχάνει τις εξής λειτουργίες:

- a) Διαβάζει από τη «standard» είσοδο ένα πλήθος αριθμών «N» στο διάστημα [6, 49].
- b) Διαβάζει από τη «standard» είσοδο «N» αριθμούς στο διάστημα [1, 49].
- c) Διαβάζει από τη «standard» είσοδο δύο αριθμούς «X1», «X2» στο διάστημα [0, X1]U[X1, X2]U[X2, 6] για το πλήθος των άρτιων αριθμών κάθε συνδυασμού.
- d) Διαβάζει από τη «standard» είσοδο δύο αριθμούς «Y1», «Y2» στο διάστημα [21, Y1]U[Y1, Y2]U[Y2, 279] για το άθροισμα των αριθμών κάθε συνδυασμού.
- e) Δημιουργεί δυναμικά δεσμευμένους πίνακες για τα δεδομένα «b», «c» και «d» αντίστοιχα.
- f) Αναζητάει στον πίνακα με τους «N» αριθμούς, αριθμούς που δεν ανήκουν στο διάστημα [1, 49] και έχουν ήδη καταχωρηθεί στον πίνακα.
- g) Ελέγχει αν υπάρχει χώρος στη μνήμη για την δέσμευση τους.
- h) Ταξινομεί τον πίνακα με τους «N» αριθμούς στο διάστημα [1, 49] κατά αύξουσα σειρά.
- i) Εκτυπώνει το πλήθος «N» των αριθμών στο διάστημα [6, 49]
- j) Εκτυπώνει τον πίνακα με τους «N» αριθμούς στο διάστημα [1, 49] ταξινομημένο κατά αύξουσα σειρά.
- k) Εκτυπώνει τους αριθμούς «X1» και «X2» του διαστήματος [0, X1]U[X1, X2]U[X2, 6] για το πλήθος των άρτιων αριθμών κάθε συνδυασμού.
- l) Εκτυπώνει τους αριθμούς «Y1» και «Y2» διαστήματος [21, Y1]U[Y1, Y2]U[Y2, 279] για το άθροισμα των αριθμών κάθε συνδυασμού.
- m) Δημιουργεί συνδυασμούς «N» ανά 6.
- n) Δημιουργεί δυναμικά δεσμευμένο πίνακα για την προσωρινή αποθήκευση κάθε συνδυασμού που δημιουργείται.
- o) Αναζητάει στον πίνακα με τον κάθε συνδυασμό για το πλήθος των άρτιων αριθμών κάθε συνδυασμού.

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

p) Αναζητάει στον πίνακα με τον κάθε συνδυασμό, το άθροισμα των αριθμών κάθε συνδυασμού.

q) Εκτυπώνει τους συνδυασμούς που το πλήθος των άρτιων αριθμών ανήκει στο διάστημα  $[X1, X2]$  και το άθροισμα των αριθμών ανήκει στο διάστημα  $[Y1, Y2]$ .

r) Υπολογίζει την συχνότητα εμφάνισης κάθε αριθμού στους εκτυπωμένους συνδυασμούς.

s) Εκτυπώνει την συχνότητα εμφάνισης κάθε αριθμού στους εκτυπωμένους συνδυασμούς.

t) Εκτυπώνει το πλήθος των συνδυασμών που πληρούν συγκεκριμένους όρους.

v) Αποδεσμεύει την μνήμη που δεσμεύτηκε δυναμικά.

## **ΔΟΜΗ**

Προκειμένου να υλοποιηθεί το ζητούμενο χρησιμοποιήθηκαν, αρχικά, οι βιβλιοθήκες (.h) και οι εντολές :

a) «stdio.h»: Περιέχει τις έτοιμες συναρτήσεις «scanf(...)» και «printf(...)» που συνδέονται με τα κανάλια εισόδου και εξόδου αντίστοιχα για την ανάγνωση και την τύπωση περιεχομένων των αντίστοιχων μεταβλητών. Επίσης, η «printf(...)» χρησιμοποιήθηκε για να τυπωθούν χαρακτηριστικά μηνύματα για την βέλτιστη κατανόηση του πηγαίου κώδικα.

b) «stdlib.h»: Περιέχει τις έτοιμες συνάρτησεις «malloc(...)» και «calloc(...)» για την δυναμική δέσμευση μνήμης για την δημιουργία δυναμικών πινάκων.

Επιπρόσθετα, χρησιμοποιήθηκαν δείκτες (τύπου «int») και οι χαρακτηριστικοί τελεστές :

a) αριθμητικοί : +

b) σχεσιακοί : <=, >=, >, <, !=, ==

c) ανάθεσης : =

d) τελεστής & : Για την διεύθυνση μεταβλητής ως δεύτερο όρισμα της συνάρτησης «scanf()» που συνδέεται με τη «standard» είσοδο.

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

e) μετααύξησης : μεταβλητή++  
Οι εντολές ελέγχου :

a) if – else

Οι εντολές επαναλήψης :

a) do – while

b) for

c) while

Οι πίνακες :

a) Δυναμικά Μονοδιάστατοι

Η κάθε λειτουργία απ' την ενότητα «Ζητούμενο» υλοποιήθηκε με αυτόνομα υποπρογράμματα (βλ. ενότητα «Συναρτήσεις»).

## **ΣΥΝΑΡΤΗΣΕΙΣ**

**Τύπου «void»** **(δεν επιστρέφουν τιμή)**

**Title ()** «Ο τίτλος του προγράμματος»

**Print\_N (int)** «Εκτύπωση του πλήθους των αριθμών στο διάστημα (6, 49]»

**Check\_Memory (int \*)** «Έλεγχος στη μνήμη για τις δυναμικές δεσμεύσεις πινάκων»

**Sort (int \*, int)** «Ταξινόμηση του πίνακα με τους αριθμούς κατά αύξουσα σειρά»

**Print\_Matrix (int \*, int)** «Εκτύπωση του πίνακα με τους "N" αριθμούς στο διάστημα [1, 49]»

**Print\_Even\_Lims (int \*)** «Σύγκριση των συναρτήσεων "Sin (ω)" και "Taylor\_S (ω)" που υπολογίζουν το ημίτονο της γωνίας σε ακτίνια για το αν είναι "σχεδόν" ίσοι»

**Print\_Sum\_Lims (int \*)** «Σύγκριση των συναρτήσεων "Cos (ω)" και "Taylor\_C (ω)" που υπολογίζουν το συνημίτονο της γωνίας σε ακτίνια για το αν είναι "σχεδόν" ίσοι»



# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

**Print\_Combos (int \*)** «Εκτύπωση των συνδυασμών»

**Frequency (int \*, int \*)** «Υπολογισμός της συχνότητας εμφάνισης κάθε αριθμού στους εκτυπωμένους συνδυασμούς»

**Print\_Freq (int \*)** «Εκτύπωση της συχνότητας εμφάνισης κάθε αριθμού στους εκτυπωμένους συνδυασμούς»

**Print\_num\_Combos (int, int \*)** «Εκτύπωση του πλήθους των συνδυασμών που πληρούν συγκεκριμένους όρους»

**Free\_Memory (int \*)** «Αποδέσμευση μνήμης»

**Τύπου «int»** **(επιστρέφουν ακέραια τιμή)**

**main (int argc, char \*\*argv)** «Η κύρια συνάρτηση του προγράμματος»

**Read\_N\_Numbers (int, int)** «Εισαγωγή του πλήθους "N" των αριθμών, όπου το "N" ανήκει στο διάστημα (6, 49]»

**Search (int \*, int, int, int, int)** «Αναζήτηση στον δυναμικά δεσμευμένο πίνακα με τους αριθμούς για πιθανόν εντοπισμό ίδιων καταχωρημένων αριθμών»

**Search\_Evens (int \*, int \*, int)** «Αναζήτηση για το πλήθος των άρτιων αριθμών κάθε συνδυασμού»

**Search\_Sum (int \*, int \*, int)** «Αναζήτηση για το άθροισμα των αριθμών κάθε συνδυασμού»

**Τύπου «int \*»** **(επιστρέφουν ακέραιο δείκτη)**

**\*Read\_Even\_Lims ()** «Εισαγωγή του διαστήματος [0, X1]U[X1, X2]U[X2, 6] για το πλήθος των άρτιων αριθμών κάθε συνδυασμού»

**\*Read\_Sum\_Lims ()** «Εισαγωγή του διαστήματος [21, Y1]U[Y1, Y2]U[Y2, 279] για το άθροισμα των αριθμών κάθε συνδυασμού»

**\*Combos (int \*, int, int \*, int \*)** «Δημιουργία των συνδυασμών»

## **METABΛΗΤΕΣ**

**main (int argc, char \*\*argv)**

Ακέραιες μεταβλητές (τύπου «int»)

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

\*Mat (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Mat» που περιέχει τους «N» αριθμούς στο διάστημα [1, 49])

N (Το πλήθος «N» των αριθμών στο διάστημα (6, 49])

\*Lim\_e (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα που δημιουργεί η συνάρτηση «Read\_Even\_Lims(...)» για το διάστημα [0, X1]U[X1, X2]U[X2, 6] του πλήθους των άρτιων αριθμών κάθε συνδυασμού)

\*Lim\_s (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα που δημιουργεί η συνάρτηση «Read\_Sum\_Lims(...)» για το διάστημα [21, Y1]U[Y1, Y2]U[Y2, 279] του αθροίσματος αριθμών κάθε συνδυασμού)

\*c (Δείκτης που δείχνει στον δυναμικά δεσμευμένο ακέραιο πίνακα που δημιουργεί η συνάρτηση «Combos(...)» και περιέχει προσωρίνα έναν έναν τους συνδυασμούς)

## **Read N Numbers (int L1, int L2)**

### Παράμετροι

L1 (Η σταθερά «6»)

L2 (Η σταθερά «49»)

### Ακέραιες μεταβλητές (τύπου «int»)

x (Το πλήθος «N» των αριθμών στο διάστημα (6, 49])

## **Print N (int N)**

### Παράμετρος

N (Το πλήθος «N» των αριθμών στο διάστημα (6, 49])

## **Read Matrix (int \*M, int N)**

### Παράμετροι

\*M (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Mat» που περιέχει τους «N» αριθμούς στο διάστημα [1, 49])

N (Το πλήθος «N» των αριθμών στο διάστημα (6, 49])

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## Ακέραιες μεταβλητές (τύπου «int»)

i (Βοηθητική μεταβλητή για τον έλεγχο του βρόχου)

## **Search (int \*M, int N, int Num, int O1, int O2)**

### Παράμετροι

\*M (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Mat» που περιέχει τους «N» αριθμούς στο διάστημα [1, 49])

N (Το πλήθος «N» των αριθμών στο διάστημα (6, 49])

Num (Ο αριθμός που διαβάστηκε από τη «standard» είσοδο)

O1 (Η σταθερά «1»)

O2 (Η σταθερά «49»)

## Ακέραιες μεταβλητές (τύπου «int»)

j (Βοηθητική μεταβλητή για τον έλεγχο του βρόχου)

## **\*Read Even Lims ()**

### Ακέραιες μεταβλητές (τύπου «int»)

\*Evens (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Evens», για το διάστημα [0, X1]U[X1, X2]U[X2, 6] του πλήθους των άρτιων αριθμών κάθε συνδυασμού)

## **\*Read Sum Lims ()**

### Ακέραιες μεταβλητές (τύπου «int»)

\*Sum (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Sum», για το διάστημα [21, Y1]U[Y1, Y2]U[Y2, 279] του αθροίσματος αριθμών κάθε συνδυασμού)

## **Check Memory (int \*Array)**

### Παράμετρος

\*Array (Δείκτης που δείχνει στην πρώτη θέση ενός δυναμικά δεσμευμένου ακέραιου πίνακα)

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## **Sort (int \*M, int N)**

### Παράμετροι

\*M (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Mat» που περιέχει τους «N» αριθμούς στο διάστημα [1, 49])

N (Το πλήθος «N» των αριθμών στο διάστημα (6, 49])

### Ακέραιες μεταβλητές (τύπου «int»)

i (Βοηθητική μεταβλητή για τον έλεγχο του πρώτου βρόχου)

j (Βοηθητική μεταβλητή για τον έλεγχο του δεύτερου βρόχου)

tmp (Βοηθητική μεταβλητή για την ταξινόμηση των στοιχείων του πίνακα)

## **Print Matrix (int \*M, int N)**

### Παράμετροι

\*M (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Mat» που περιέχει τους «N» αριθμούς στο διάστημα [1, 49])

N (Το πλήθος «N» των αριθμών στο διάστημα (6, 49])

### Ακέραιες μεταβλητές (τύπου «int»)

i (Βοηθητική μεταβλητή για τον έλεγχο του βρόχου)

## **Print Even Lims (int \*E)**

### Παράμετρος

\*E (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Evens» για το διάστημα [0, X1]U[X1, X2]U[X2, 6] του πλήθους των άρτιων αριθμών κάθε συνδυασμού)

### Ακέραιες μεταβλητές (τύπου «int»)

i (Βοηθητική μεταβλητή για τον έλεγχο του βρόχου)

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## **Print Sum Lims (int \*S)**

### Παράμετρος

\*S (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Sum» για το διάστημα [21, Y1]U[Y1, Y2]U[Y2, 279] του αθροίσματος αριθμών κάθε συνδυασμού)

### Ακέραιες μεταβλητές (τύπου «int»)

j (Βοηθητική μεταβλητή για τον έλεγχο του βρόχου)

## **\*Combos (int \*M, int N, int \*E, int \*S)**

### Παράμετροι

\*M (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Mat» που περιέχει τους «N» αριθμούς στο διάστημα [1, 49])

N (Το πλήθος «N» των αριθμών στο διάστημα (6, 49])

\*E (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Evens» για το διάστημα [0, X1]U[X1, X2]U[X2, 6] του πλήθους των άρτιων αριθμών κάθε συνδυασμού)

\*S (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Sum» για το διάστημα [21, Y1]U[Y1, Y2]U[Y2, 279] του αθροίσματος αριθμών κάθε συνδυασμού)

### Ακέραιες μεταβλητές (τύπου «int»)

i (Βοηθητική μεταβλητή για τον έλεγχο του πρώτου βρόχου)

j (Βοηθητική μεταβλητή για τον έλεγχο του δεύτερου βρόχου)

k (Βοηθητική μεταβλητή για τον έλεγχο του τρίτου βρόχου)

l (Βοηθητική μεταβλητή για τον έλεγχο του τέταρτου βρόχου)

m (Βοηθητική μεταβλητή για τον έλεγχο του πέμπτου βρόχου)

n (Βοηθητική μεταβλητή για τον έλεγχο του έκτου βρόχου)

evens (Η τιμή επιβεβαίωσης που επιστρέφει η συνάρτηση «Search\_Evens(...)»

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

για το πλήθος των άρτιων αριθμών κάθε συνδυασμού)

sum (Η τιμή επιβεβαίωσης που επιστρέφει η συνάρτηση «Search\_Sum(...)» για το άθροισμα των αριθμών κάθε συνδυασμού)

Ακέραιες μεταβλητές δείκτη (τύπου «int \*»)

\*Com (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Com» με τον κάθε συνδυασμό που δημιουργείται, προσωρινά)

\*Cnt (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Cnt» με τους κατάλληλους μετρητές, για τον υπολογισμό του πλήθους συνδυασμών που πληρούν συγκεκριμένους όρους)

\*Freq (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Freq» με τους κατάλληλους μετρητές, για τον υπολογισμό της συχνότητας εμφάνισης κάθε «N» αριθμού στους συνδυασμούς που τυπώθηκαν)

## Search Evens (int \*C, int \*E, int N)

### Παράμετροι

\*C (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Com» με τον κάθε συνδυασμό που δημιουργείται, προσωρινά)

N (Το πλήθος «N» των αριθμών στο διάστημα (6, 49])

\*E (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Evens» για το διάστημα [0, X1]U[X1, X2]U[X2, 6] του πλήθους των άρτιων αριθμών κάθε συνδυασμού)

Ακέραιες μεταβλητές (τύπου «int»)

j (Βοηθητική μεταβλητή για τον έλεγχο του βρόχου)

evens (Η τιμή επιβεβαίωσης για το πλήθος των άρτιων αριθμών κάθε συνδυασμού)

## Search\_Sum (int \*C, int \*S, int N)

### Παράμετροι

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

\*C (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Com» με τον κάθε συνδυασμό που δημιουργείται, προσωρινά)

N (Το πλήθος «N» των αριθμών στο διάστημα (6, 49])

\*S (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Sum» για το διάστημα [21, Y1]U[Y1, Y2]U[Y2, 279] του αθροίσματος αριθμών κάθε συνδυασμού)

Ακέραιες μεταβλητές (τύπου «int»)

j (Βοηθητική μεταβλητή για τον έλεγχο του βρόχου)

sum (Η τιμή επιβεβαίωσης για το άθροισμα των αριθμών κάθε συνδυασμού)

**Print Combos (int \*C)**

Παράμετρος

\*C (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Com» με τον κάθε συνδυασμό που δημιουργείται, προσωρινά)

**Frequency (int \*F, int \*C)**

Παράμετροι

\*F (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Freq» με τους μετρητές συχνότητας εμφάνισης κάθε «N» αριθμού στους συνδυασμούς που τυπώθηκαν)

\*C (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Com» με τον κάθε συνδυασμό που δημιουργείται, προσωρινά)

**Print Freq (int \*F)**

Παράμετρος

\*F (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Freq» με τους μετρητές συχνότητας εμφάνισης κάθε «N» αριθμού στους συνδυασμούς που τυπώθηκαν)

Ακέραιες μεταβλητές (τύπου «int»)

i (Βοηθητική μεταβλητή για τον έλεγχο του βρόχου)

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## **Print\_num\_Combos (int N, int \*Cnt)**

### Παράμετρος

\*Cnt (Δείκτης που δείχνει στην πρώτη θέση του δυναμικά δεσμευμένου ακέραιου πίνακα «Cnt» με τους κατάλληλους μετρητές, για τον υπολογισμό του πλήθους συνδυασμών που πληρούν συγκεκριμένους όρους)

## **Free\_Memory (int \*Array)**

### Παράμετρος

\*Array (Δείκτης που δείχνει στην πρώτη θέση ενός δυναμικά δεσμευμένου ακέραιου πίνακα)

## **ΔΙΑΣΧΙΣΗ**

Βλ. Σχόλια «Πρόγραμμα “Combinations.c”»

## **ΠΑΡΑΔΕΙΓΜΑΤΑ**

### **Παράδειγμα 1**

Στοιχείο [ 1] Τιμή : 2

Στοιχείο [ 2] Τιμή : 6

Στοιχείο [ 3] Τιμή : 7

Στοιχείο [ 4] Τιμή : 12

Στοιχείο [ 5] Τιμή : 21

Στοιχείο [ 6] Τιμή : 32

Στοιχείο [ 7] Τιμή : 43

Στοιχείο [ 8] Τιμή : 45

N : 8

X1 : 2

X2 : 4



# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Y1 : 21

Y2 : 141

2 6 7 12 21 32

2 6 7 12 21 43

2 6 7 12 21 45

2 6 7 12 32 43

2 6 7 12 32 45

2 6 7 12 43 45

2 6 7 21 32 43

2 6 7 21 32 45

2 6 7 21 43 45

2 6 7 32 43 45

2 6 12 21 32 43

2 6 12 21 32 45

2 6 12 21 43 45

2 6 12 32 43 45

2 7 12 21 32 43

2 7 12 21 32 45

2 7 12 21 43 45

2 7 12 32 43 45

6 7 12 21 32 43

6 7 12 21 32 45

6 7 12 21 43 45

Το πλήθος των συνδυασμών N ανά 6 είναι : 28

Το πλήθος των συνδυασμών που δεν πληρούσαν τον πρώτο όρο : 0

Το πλήθος των συνδυασμών που πληρούσαν τον πρώτο, αλλά δεν πληρούσαν τον δεύτερο όρο : 7

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Το πλήθος των συνδυασμών που τυπώθηκαν  
21

:

Συχνότητα του [ 2] :	18
Συχνότητα του [ 6] :	17
Συχνότητα του [ 7] :	17
Συχνότητα του [12] :	17
Συχνότητα του [21] :	15
Συχνότητα του [32] :	14
Συχνότητα του [43] :	14
Συχνότητα του [45] :	14

## **ΠΑΡΑΤΗΡΗΣΕΙΣ**

Τα όρια του αθροίσματος  $[21, Y1] \cup [Y1, Y2] \cup [Y2, 279]$  των αριθμών ενός συνδυασμού που πρέπει να είναι για να είναι ένας συνδυασμός υποψήφιος για να τυπωθεί, έχουν μία λογική εξήγηση.

Ο «μικρότερος» συνδυασμός ανά «6» που μπορεί να δημιουργηθεί από αριθμούς του διαστήματος  $[1, 49]$  είναι ο «1, 2, 3, 4, 5, 6». Αν τους προσθέσουμε, το άθροισμα τους είναι ίσο με την τιμή «21», οπότε και ορίζεται ως το ελάχιστο όριο αθροίσματος ενός συνδυασμού ανά «6» με αριθμούς του διαστήματος  $[1, 49]$ .

Από την άλλη, ο «μεγαλύτερος» συνδυασμός ανά «6» που μπορεί να δημιουργηθεί από αριθμούς του διαστήματος  $[1, 49]$  είναι ο «44, 45, 46, 47, 48, 49». Αν τους προσθέσουμε, το άθροισμα τους είναι ίσο με την τιμή «279», οπότε και ορίζεται ως το μέγιστο όριο αθροίσματος ενός συνδυασμού ανά «6» με αριθμούς του διαστήματος  $[1, 49]$ .

