

6η Εργασία – Παίζουμε τριανταμία;

Ανάλυση του προβλήματος

Το παιχνίδι θα παίζεται με μια στοίβα η οποία μπορεί να περιέχει κάρτες από μία ή περισσότερες τράπουλες. Θα προσομοιάσουμε τη στοίβα με τις κάρτες με μια δομή στοίβας. Η δομή των κόμβων της στοίβας θα περιέχει πληροφορίες για το είδος της κάρτας (καρό, κούπα, σπαθί, μπαστούνι), τη φιγούρα (π.χ. Άσος, Βαλές, πέντε, κ.λπ.) και την αξία σε πόντους.

Αρχικά στη στοίβα προστίθενται κάρτες από μία ή περισσότερες τράπουλες. Στη συνέχεια τα περιεχόμενα της στοίβας ανακατεύονται ώστε η σειρά τους να είναι τυχαία.

Όταν ξεκινά το παιχνίδι, τόσο η «μάννα» όσο και ο παίκτης τραβούν κάρτες από την κορυφή της στοίβας, σύμφωνα με τις οδηγίες της εκφώνησης. Κάθε κάρτα που χρησιμοποιείται απωθείται από τη στοίβα.



Βήματα

- 1 Το πρώτο βήμα είναι να δημιουργήσουμε τα απαραίτητα στοιχεία για τη διαχείριση μιας στοίβας, η οποία θα περιέχει κόμβους που αντιστοιχούν σε κάρτες τράπουλας. Θα χρησιμοποιηθεί η ίδια προσέγγιση με αυτήν του προγράμματος 18_new_stack.c του βιβλίου. Η δομή των κόμβων της στοίβας `stack_node` πρέπει να διαθέτει πεδία για την καταχώριση του είδους της κάρτας, της φιγούρας της και της αξίας της. Η διαχείριση της στοίβας γίνεται με τις συναρτήσεις `create_stack()` για τη δημιουργία μιας στοίβας, `push_in_stack()` για την ώθηση ενός κόμβου, και `pop_from_stack()` για την απώθηση ενός κόμβου από τη στοίβα.
- 2 Δημιουργούμε τη συνάρτηση `fill()` η οποία συμπληρώνει τη στοίβα με κάρτες από μία ή περισσότερες τράπουλες. Στη συνάρτηση μεταβιβάζεται ένας δείκτης (`tr`) προς τη στοίβα, στην οποία θα προστεθούν οι κάρτες της τράπουλας, καθώς και το πλήθος (`ar`) των τραπουλών που θέλουμε να προσθέσουμε. Η συνάρτηση δημιουργεί όλους τους δυνατούς συνδυασμούς που υπάρχουν σε κάθε τράπουλα, καταχωρίζοντας τα στοιχεία αυτά στα πεδία μιας δομής τύπου `stack_node` και στη συνέχεια, χρησιμοποιώντας τη συνάρτηση `push_in_stack()`, προσθέτει έναν κόμβο με αυτά τα στοιχεία στην κορυφή της στοίβας.
- 3 Το επόμενο βήμα είναι να δημιουργήσουμε τη συνάρτηση `anakatema()` η οποία θα ανακατεύει τις κάρτες της στοίβας με κάποιο τυχαίο τρόπο, διαφορετικά θα ήταν πάντα τοποθετημένες με την ίδια σειρά. Στη συνάρτηση μεταβιβάζεται ένας δείκτης προς τη στοίβα, καθώς και το πλήθος των καρτών που περιέχει. Η τεχνική βάσει της οποίας γίνεται το ανακάτεμα είναι η εξής: Επιλέγουμε δύο τυχαίες κάρτες από τη στοίβα και τις αντιμεταθέτουμε. Αν αυτό γίνει πολλές φορές οι κάρτες της στοίβας θα ανακατευτούν και θα τοποθετηθούν σε τυχαίες θέσεις. Η επιλογή των τυχαίων καρτών γίνεται με χρήση της συνάρτησης βιβλιοθήκης `rand()` και η αντιμετάθεσή τους με τη συνάρτηση `swap_data()` που θα δημιουργήσουμε για αυτόν τον σκοπό.
- 4 Η καρδιά του προγράμματος θα αποτελείται από τη συνάρτηση `rexe_31()`, η οποία υλοποιεί τη λειτουργία του παιχνιδιού, σύμφωνα με τις οδηγίες της εκφώνησης. Στη συνάρτηση μεταβιβάζεται ένας δείκτης προς τη στοίβα με τις κάρτες, καθώς και το πλήθος των καρτών που περιέχει. Αρχικά ο παίκτης τραβά μία κάρτα από τη στοίβα και ποντάρει το ποσό που επιθυμεί. Η κάρτα αυτή απωθείται από τη στοίβα.
- 5 Έπειτα θα είναι η σειρά της «μάννας» να τραβήξει χαρτιά. Η «μάννα» τραβάει χαρτιά μέχρι να βγάλει άθροισμα πόντων μεγαλύτερο από 23, ίσο με 14 ή 31, ή μέχρι να καεί (περισσότεροι από 31 πόντοι). Αν συμπληρώσει 31 πόντους κερδίζει το ποντάρισμα ενώ αν «καεί» το χάνει.

- ❖ Αν δεν κερδίσει ή δεν χάσει η «μάννα», είναι σειρά του παίκτη να τραβήξει κάρτες από τη στοίβα. Τραβάει κάρτες μέχρις ότου αποφασίσει να σταματήσει, να συμπληρώσει 31 πόντους οπότε κερδίζει, ή να «καεί» οπότε χάνει.
- ❖ Αν κανένας από τους δύο δεν έχει κερδίσει ή δεν έχει καεί, τότε συγκρίνονται οι πόντοι των δύο παικτών και αποφασίζεται ποιος από τους δύο κερδίζει.
- ❖ Μετά εμφανίζεται το συνολικό ποσό που κερδίζει ή χάνει ο παίκτης και το παιχνίδι συνεχίζεται με τον επόμενο γύρο. Αν δεν έχουν μείνει αρκετές κάρτες στη στοίβα για ένα νέο γύρο, ζητείται από τον παίκτη να αποφασίσει αν θα προστεθούν στη στοίβα κάρτες από μία ακόμα τράπουλα. Σε αυτή την περίπτωση, η στοίβα συμπληρώνεται με τη νέα τράπουλα, η οποία ανακατεύεται, και το παιχνίδι συνεχίζεται, διαφορετικά το παιχνίδι σταματά.
- ❖ Στη συνάρτηση `main()` μένει να υλοποιήσουμε ελάχιστα πράγματα. Αρχικά δημιουργούμε τη στοίβα, τη συμπληρώνουμε με κάρτες και την ανακατεύουμε, χρησιμοποιώντας τις αντίστοιχες συναρτήσεις που έχουμε ήδη περιγράψει. Τέλος, καλούμε τη συνάρτηση `pexe_31()` η οποία κάνει όλη τη δουλειά!

Κώδικας

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <string.h>
#include <time.h>
struct stack_node
{
    int axia;
    char eidos[10];
    char figoyra[10];
    struct stack_node *next;
};
struct stack
{
    struct stack_node *top;
};
void pop_from_stack(struct stack *s);
bool push_in_stack(struct stack *s, struct stack_node x);
bool is_stack_empty(struct stack *s);
struct stack *create_stack();
void display_stack(struct stack *s);
void fill(struct stack *tr, int ar);
void anakatema(struct stack *tr, int kartes);
void swap_data(struct stack_node *p1, struct stack_node *p2);
void pexe_31(struct stack *tr, int kartes);

int main(void)
{
    char ch;
    int timi, artr, kartes;
    struct stack *trapouyla=create_stack();
    printf("Πόσες τράπουλες θα έχει η στοίβα :");
```

e6.c

❖ Η δομή των κόμβων της στοίβας. Κάθε κόμβος αντιστοιχεί σε μια κάρτα της τράπουλας. Περιέχει την αξία της, το είδος της (καρό, κόυπα, σπαθί, μπαστούνι) και τη φιγούρα της (π.χ. Άσος, Ντάμα, Οκτώ, κ.λπ.). Επίσης, περιέχει έναν δείκτη προς τον επόμενο κόμβο.

❖ Η δομή **stack** προσδιορίζει μια στοίβα. Περιέχει μόνο ένα πεδίο, έναν δείκτη προς τον κόμβο κορυφής της.

Προκαταβολικές δηλώσεις των συναρτήσεων

❖ Δημιουργούμε μια στοίβα την οποία ονομάζουμε **trapouyla**. Το αναγνωριστικό **trapouyla** είναι ένας δείκτης προς τον κόμβο κορυφής της στοίβας. Μέσω αυτού του δείκτη θα διαχειριζόμαστε τη στοίβα.

```
scanf("%d",&artr);
kartes=artr*52;
fill(trapoyla,artr);
anakatema(trapoyla,kartes);
puts("=====");
pexe_31(trapoyla, kartes);
return 0;
}
```

➊ Συμπληρώνουμε τη στοίβα **trapoyla** με κάρτες από τόσες τράπουλες όσες έχει δηλώσει ο χρήστης (**artr**).

➋ Ανακατεύουμε τη στοίβα **trapoyla** η οποία αποτελείται από **kartes** πλήθος καρτών.

➌ Καλούμε τη συνάρτηση **pexe_31()** με την οποία παίζουμε το παιχνίδι, χρησιμοποιώντας τη στοίβα **trapoyla** η οποία περιέχει **kartes** πλήθος καρτών.

```
struct stack *create_stack()
{
    struct stack *st=(struct stack*) malloc(sizeof(struct stack));;
    st->top=NULL;
    return st;
};
```

➍ Η συνάρτηση **create_stack()** δημιουργεί μια κενή στοίβα και επιστρέφει ως τιμή έναν δείκτη προς την κορυφή της.

➎ Τοποθετεί (ωθεί) στην κορυφή της στοίβας **s** τον κόμβο **x**.

```
bool push_in_stack(struct stack *s, struct stack_node x)
{
    struct stack_node *neos;
    neos=(struct stack_node *)malloc(sizeof(struct stack_node));
    if (neos==NULL) return false;
    *neos=x;
    neos->next=s->top;
    s->top=neos;
    return true;
}
```

➏ Απωθεί έναν κόμβο από την κορυφή της στοίβας **s**.

```
void pop_from_stack(struct stack *s)
{
    struct stack_node *temp;
    int d;
    if (is_stack_empty(s))
    {
        puts("Κενή στοίβα");
        return;
    }
    temp=s->top->next;
    free(s->top);
    s->top=temp;
}
```

➐ Ελέγχει αν η στοίβα **s** είναι κενή.

```
bool is_stack_empty(struct stack *s)
{
    if (s->top==NULL)
        return true;
    else
        return false;
}
```

```
void display_stack(struct stack *s)
{
    struct stack_node *p;
    if (is_stack_empty(s))
        return;
    p=s->top;
    while (p)
    {
        printf("%s %s\n",p->figoyra,p->eidoidos);
        p=p->next;
    }
    printf("\n");
}
```

Εμφανίζει τις κάρτες της στοίβας **s**. Η συνάρτηση δεν χρησιμοποιείται στο πρόγραμμα, εκτός βέβαια αν θέλουμε να κλέβουμε!

```
void fill(struct stack *tr, int ar)
{
```

❷ Συμπληρώνει τη στοίβα που της μεταβιβάζεται στην παράμετρο **tr**, με κάρτες από ένα πλήθος τραπουλών που μεταβιβάζεται στην παράμετρο **ar**.

```
    struct stack_node karta;
    int i,j,t;
    char eidei[4][10]={"Κούπα","Καρό","Σπαθί","Μπαστούνι"};
    char figoyres[13][10]={"Άσος","Δύο","Τρία","Τέσσερα","Πέντε","Έξι","Εφτά","Οκτώ",
        "Εννιά","Δέκα","Βαλές","Ντάμα","Ρήγας"};
    int axes[13]={1,2,3,4,5,6,7,8,9,10,10,10,10};
    for (t=1;t<=ar;t++)
        for (i=0;i<4;i++)
            for (j=0;j<13;j++)
            {
                karta.axia=axes[j];
                strcpy(karta.eidoidos,eidei[i]);
                strcpy(karta.figoyra,figoyres[j]);
                push_in_stack(tr,karta);
            }
}
```

❷ Στα πεδία της μεταβλητής δομής **karta** καταχωρίζονται μία προς μία οι κάρτες που διαθέτει μια τράπουλα. Αυτό γίνεται 52 φορές (4x13). Στην περίπτωση που προστεθούν παραπάνω τράπουλες (**ar**>1), τότε η διαδικασία επαναλαμβάνεται ισάριθμες (**ar**) φορές.

```
void anakatema(struct stack *tr, int kartes)
{
```

❸ Ανακατεύει τις κάρτες μιας στοίβας που της μεταβιβάζεται στην παράμετρο **tr**. Το πλήθος των καρτών της στοίβας μεταβιβάζεται στην παράμετρο **kartes**.

```
    struct stack_node *x1,*x2,*p,temp;
    int i,ta1,ta2,aa,tt;
    srand(time(NULL));
    for (i=1;i<=kartes*20;i++)
    {
        aa=0;
        ta1=rand()%kartes+1;
        ta2=rand()%kartes+1;
        if (ta1>ta2)
        {
            tt=ta1;
            ta1=ta2;
            ta2=tt;
        }
    }
```

Η τεχνική με την οποία γίνεται το ανακάτεμα είναι η ακόλουθη: Δημιουργούμε δύο τυχαίους αριθμούς από το 1 μέχρι το πλήθος των καρτών της στοίβας. Έστω, για παράδειγμα, ότι είναι το 10 και το 45. Διασχίζουμε τη στοίβα και εντοπίζουμε τον 10ο και τον 45ο κόμβο, που αντιστοιχούν σε αυτές τις κάρτες, και αντιμεταθέτουμε τα περιεχόμενα αυτών των κόμβων. Όσο περισσότερες φορές κάνουμε αυτή τη διαδικασία τόσο καλύτερα θα ανακατευτούν οι κάρτες της στοίβας. Στην περίπτωση μας γίνεται τόσες φορές όσο είναι το πλήθος των καρτών επί 20! Δηλαδή, αν η στοίβα διαθέτει 52 κάρτες, θα γίνουν 1040 (20*52) αντιμεταθέσεις.

```

    p=tr->top;
    while (p)
    {
        aa++;
        if (aa==ta1)
            x1=p;
        else if (aa==ta2)
        {
            x2=p;
            break;
        }
        p=p->next;
    }
    swap_data(x1,x2);
}

```

❷ Καλεί τη συνάρτηση **swap_data()** για την αντιμετάθεση των περιεχομένων των κόμβων στους οποίους δείχνουν οι δείκτες **x1** και **x2**.

❸ Η συνάρτηση **swap_data()** αντιμεταθέτει τα περιεχόμενα δύο κόμβων τύπου **stack_node**, δείκτες προς τους οποίους μεταβιβάζονται στις παραμέτρους **p1** και **p2**.

```

void swap_data(struct stack_node *p1, struct stack_node *p2)
{
    struct stack_node *p1_next, *p2_next, temp;
    p1_next=p1->next;
    p2_next=p2->next;
    temp=*p1;
    *p1=*p2;
    *p2=temp;
    p1->next=p1_next;
    p2->next=p2_next;
}

```

❹ Η συνάρτηση **pexe_31()** υλοποιεί τη λογική του παιχνιδιού της τριανταμίας. Στην παράμετρο **tr** μεταβιβάζεται η στοίβα με τις κάρτες και στην παράμετρο **kartes** μεταβιβάζεται το πλήθος των καρτών που περιέχει.

```

void pexe_31(struct stack *tr, int kartes)
{
    struct stack_node kartes_paikti[10],kartes_manas[10];
    int xp,xm,bp,bm,poso,synolo;
    int gyroi=1,perasan=0;
    char ans;
    do
    {
        xp=xm=0;
        bp=bm=0;
        printf("\n##### ΤΥΠΟΣ Νο %d - Κάρτες στη στοίβα %d #####\n\n",
            ,gyroi,kartes-perasan);
        kartes_paikti[xp]=*(tr->top);
        pop_from_stack(tr);
        perasan++;
        bp=bp+kartes_paikti[xp].axia;
        printf("Κάρτα παίκτη : %s %s Σύνολο πόντων : %d\n",
            kartes_paikti[xp].figoyra,kartes_paikti[xp].eidon,bp);
        xp++;
    }
}

```

❺ Ο παίκτης τραβάει μια κάρτα από την κορυφή της στοίβας. Η κάρτα αυτή στη συνέχεια απομακρύνεται από τη στοίβα.

```
printf("Δώσε το ποντάρισμα σου (1~10 euro) -> ");
scanf("%d",&poso);
printf("\nΤώρα τραβάει η μάνα\n");
while (bm<=23)
{
    kartes_manas[xm]=*(tr->top);
    bm=bm+kartes_manas[xm].axia;
    xm++;
    pop_from_stack(tr);
    perasan++;
    if (bm==14) break;
}
if (bm==31)
{
    printf("Τριανταμία ... σε κέρδισα\n");
    synolo=synolo+poso;
}
else if (bm>31)
{
    printf("Ωπ... κάηκα %d\n",bm);
    synolo=synolo+poso;
}
else
{
    printf("Έχω %d πόντους\n",bm);
    printf("\nΤώρα τραβάει ο παίκτης \n");
    do
    {
        kartes_paikti[xp]=*(tr->top);
        pop_from_stack(tr);
        perasan++;
        bp=bp+kartes_paikti[xp].axia;
        printf("Κάρτα παίκτη : %s %s Σύνολο πόντων : %d\n",
            kartes_paikti[xp].figoyra,kartes_paikti[xp].eidos,bp);
        xp++;
        if (bp>=31) break;
        printf("\nΠάτησε ENTER για επόμενη κάρτα - οτιδήποτε άλλο για διακοπή >");
        fflush(stdin);
        ans=getchar();
    } while (ans!='\n');
    if (bp==31)
    {
        printf("Μπράβο ΤΡΙΑΝΤΑΜΙΑ κέρδισες!!!\n",bp);
        synolo=synolo+poso;
    }
    else if (bp>31)
    {
        printf("Δυστυχώς κάηκες %d\n",bp);
    }
}
```

❶ Ζητείται από τον παίκτη να ποντάρει ένα ποσό.

❷ Είναι η σειρά της «μάνας» να τραβήξει κάρτες από τη στοίβα. Η μάνα είναι υποχρεωμένη να τραβάει κάρτες αν έχει σύνολο πόντων κάτω από 24, εκτός αν έχει πετύχει 14.

❸ Αν συμπληρώσει 14 πόντους σταματά να τραβάει κάρτες.

❹ Αν η «μάνα» συμπληρώσει 31 πόντους κερδίζει και ο παίκτης χάνει το ποσό.

❺ Αν η «μάνα» συμπληρώσει περισσότερους από 31 πόντους «καίγεται»,σταματά να τραβάει κάρτες, και ο παίκτης κερδίζει!

❻ Σε κάθε άλλη περίπτωση είναι σειρά του παίκτη να τραβήξει κάρτες.

❼ Ο παίκτης σταματά να τραβάει κάρτες όταν συμπληρώσει σύνολο πόντων >=31

❽ Αν ο παίκτης έχει συμπληρώσει ακριβώς 31 πόντους κερδίζει!

❾ Αν ο παίκτης έχει περισσότερους από 31 πόντους καίγεται και χάνει το ποσό!

```

        synolo=synolo-poso;
    }
    else if (bm==14)
    {
        printf("Δυστυχώς σε κέρδισα γιατί έχω 14\n",bp,bm);
        synolo=synolo-poso;
    }
    else if (bp==14)
    {
        printf("Μπράβο με κέρδισες έχεις 14\n",bp,bm);
        synolo=synolo+poso;
    }
    else if (bp>bm)
    {
        printf("Μπράβο με κέρδισες έχεις %d και έχω %d πόντους\n",bp,bm);
        synolo=synolo+poso;
    }
    else if (bp==bm)
    {
        printf("Έχουμε και οι δύο %d πόντους αλλά είμαι μάνα και σε κεδρίζω!")\n",bp);
        synolo=synolo-poso;
    }
    else
    {
        printf("Δυστυχώς σε κέρδισα έχεις %d και έχω %d πόντους\n",bp,bm);
        synolo=synolo-poso;
    }
}
if (synolo>0)
    printf("\n ***** Μέχρι τώρα κερδίζεις %d euro\n",synolo);
else if (synolo<0)
    printf("\n ***** Μέχρι τώρα χάνεις %d euro\n",-synolo);
else
    printf("\n ***** Μέχρι τώρα δεν κερδίζεις ούτε χάνεις\n");
gyroi++;
if (kartes-perasan<=10)
{
    printf("Δεν μπορούμε να παίξουμε άλλο γύρο - έμειναν μόνο %d κάρτες
           στη στοίβα\n",kartes-perasan);
    printf("Θέλετε να προστεθεί στη στοίβα άλλη μια τράπουλα; N/O >");
    fflush(stdin);
    ans=getchar();
    if (ans=='N' || ans=='n')
    {
        fill(tr,1);
        kartes=kartes-perasan+52;
        perasan=0;
        anakatema(tr,kartes);
    }
}

```

7 Αν η «μάννα» έχει 14 κερδίζει!

7 Αν ο παίκτης έχει 14 κερδίζει!

7 Αν ο παίκτης έχει περισσότερους πόντους από τη «μάννα» κερδίζει.

7 Αν ο παίκτης έχει ίδιους πόντους με τη «μάννα» χάνει.

8 Εμφανίζονται τα συνολικά αποτελέσματα των πονταρισμάτων.

8 Αν στη στοίβα έχουν μείνει λίγες κάρτες, δεν μπορεί να συνεχιστεί το παιχνίδι και προτείνεται να προστεθεί ακόμα μια τράπουλα στη στοίβα.

8 Αν ο παίκτης απαντήσει θετικά, προστίθεται ακόμα μια τράπουλα στη στοίβα και στη συνέχεια η στοίβα ανακατεύεται ξανά.

```
    }  
    else break;  
}  
printf("\nΠάτησε ENTER για επόμενο γύρο - οτιδήποτε άλλο για να σταματήσεις >");  
fflush(stdin);  
ans=getchar();  
} while (ans=='\n');  
if (synolo>0)  
    printf("\nΜΠΡΑΒΟ κερδίζεις συνολικά %d euro\n",synolo);  
else if (synolo<0)  
    printf("\nΔΥΣΤΥΧΩΣ έχασες συνολικά %d euro\n",-synolo);  
else  
    printf("\nΟύτε γάτα ούτε ζημιά, δεν κέρδισες αλλά δεν έχασες κιόλας\n");  
}
```

Προτάσεις

Στο παραπάνω πρόγραμμα, κάθε Άσος έχει πάντα αξία ενός πόντου! Όμως, στο πραγματικό παιχνίδι της τριανταμίας οι Άσοι μετράνε είτε για 1 είτε για 11, ανάλογα με την επιλογή του παίκτη. Σε περίπτωση περισσότερων Άσων (2, 3, ή 4), τότε ο ένας μετράει υποχρεωτικά για 11. Προσπαθήστε να ενσωματώσετε στο πρόγραμμα αυτή τη δυνατότητα, όμως σας προειδοποιώ: δεν είναι πολύ εύκολο!

Επίσης, με βάση τη στοιβία με τις κάρτες του παραπάνω προγράμματος, μπορούμε να υλοποιήσουμε και άλλα παιχνίδια τράπουλας, όπως το blackjack ή το poker.