

INFORMATIČKI KLUB **FUTURA**

```
#include<stdio.h> C  
int main()  
{ RADIONICE PROGRAMIRANJA  
  printf("Hello World!");  
  ZA SREDNJE ŠKOLE  
  return 0;  
}
```

RADIONICE PROGRAMIRANJA ZA SREDNJE ŠKOLE - 5. RADIONICA

Nikola Rabrenović i Krunoslav Žubrinić
Informatički klub FUTURA
Dubrovnik, 10. siječnja 2015.



Creative Commons



☐ **slobodno smijete:**

- **dijeliti** — umnožavati, distribuirati i javnosti priopćavati djelo
- **remiksirati** — prerađivati djelo



☐ **pod sljedećim uvjetima:**

- **imenovanje.** Morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
- **nekomercijalno.** Ovo djelo ne smijete koristiti u komercijalne svrhe.
- **dijeli pod istim uvjetima.** Ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod licencom koja je ista ili slična ovoj.



U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela. Najbolji način da to učinite je linkom na ovu internetsku stranicu.

Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.

Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.

Tekst licence preuzet je s <http://creativecommons.org/>.

Sadržaj

- ☐ Rješenje prethodnog izazova
- ☐ Priprema za natjecanje
 - RGB
 - Skokovi
 - Sortiranje
 - ☐ Sortiranje biranjem
 - ☐ Bubble sort
 - Kirk's Quest - The Descent

Prošli 2-tjedni izazov



Codingame: Temperatures

<http://www.codingame.com/ide/7660576178af82d41e26a75fac2484e5153c88>

Trebate analizirati zapis o temperaturama i pronaći temperaturu koja je najbliža nuli.

ULAZNI PODACI:

- ☐ N (cijeli broj) broj (broj temperatura) $0 \leq N < 10000$
- ☐ Niz znakova koji sadrži N temperatura (prikazanih cijelim brojevima u intervalu između -273 i 5526).

IZLAZNI PODACI:

- ☐ 0 (nula), ako nije ponuđena niti jedna temperatura.
- ☐ U suprotnom treba prikazati **temperaturu najbližu nuli**. Ako su dva broja jednako udaljena od nule, a imaju iste predznake, bliža je pozitivna temperatura.

ULAZ 5 "1 -2 -8 4 5"	ULAZ 10 "-5 -4 -2 12 -40 4 2 18 11 5"	ULAZ 0 ""	ULAZ 1 "-2573"	ULAZ 1 "125"
IZLAZ 1	IZLAZ 2	IZLAZ 0	IZLAZ -2573	IZLAZ 125

Rješenje prošlog 2-tjednog izazova



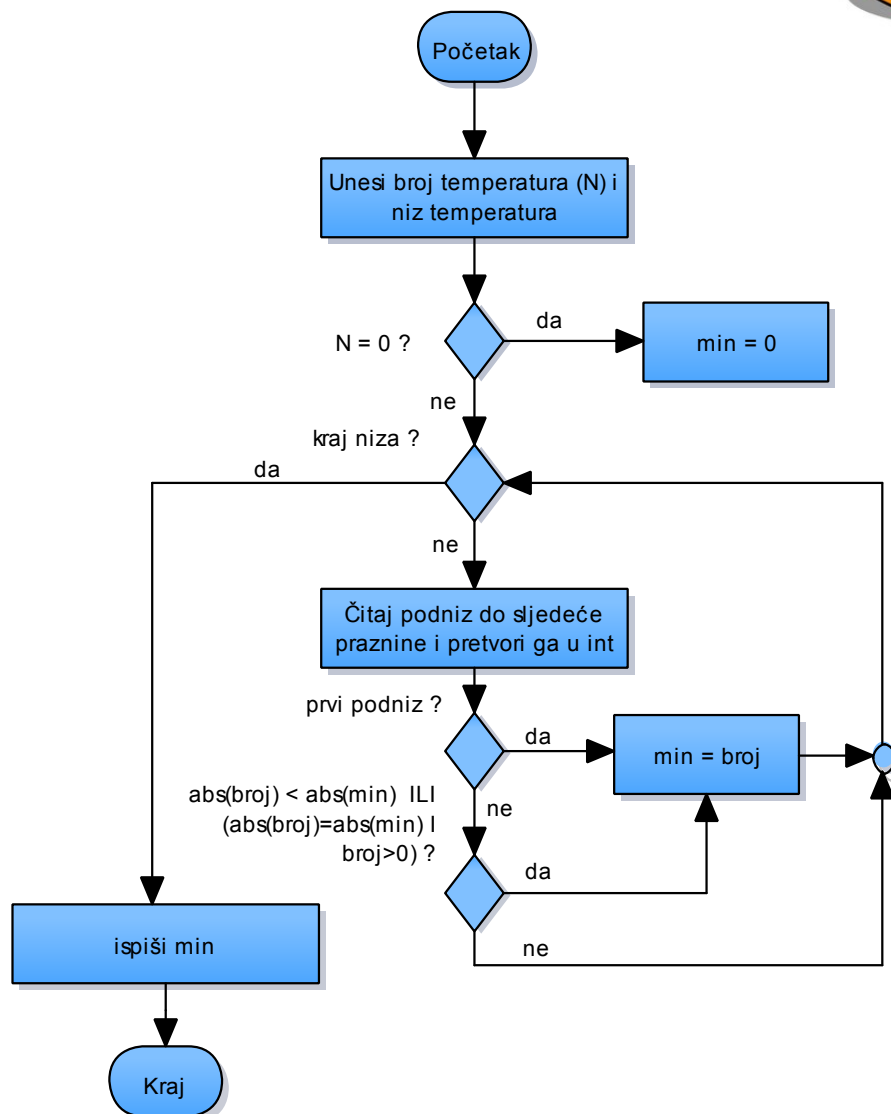
Algoritam



Rezultat
izvođenja



```
10  
-5 -4 -2 12 -40 4 2 18 11 5  
2
```



Rješenje – funkcija koja vraća min

```
int vratiMin(int pom, int min){  
    if (abs(pom)<abs(min)){  
        min = pom;  
    }  
    else if ((abs(pom) == abs(min)) &&  
        (pom>0)){  
        min = pom;  
    }  
    return min;  
}
```

Rješenje – glavni dio programa 1/2

```
int main()
{
    int N, i = 0, pom, min = 0, i1, i2 = -1;
    char TEMPS[256], poms[5];
    scanf("%d", &N); fgetc(stdin);
    fgets(TEMPS, 256, stdin);
    // ako nije unesena niti jedna temperatura – vraća 0
    if (N == 0){
        min = 0;
    }
    else {
        i1 = 0;
        ...
    }
}
```

Rješenje – glavni dio programa 2/2

```
while (TEMPS[i] != '\0'){
    if ((TEMPS[i] == ' ') || (TEMPS[i] == '\n')){
        poms[i1] = '\0';
        pom = atoi(poms);
        poms[0] = '\0';
        i1 = 0;
        if (i2 == 0) min = vratiMin(pom, min);
        else{
            min = pom;
            i2 = 0;
        }
    }
    else {
        poms[i1] = TEMPS[i];
        i1++;
    }
    i++;
}
printf("%d\n", min);
return 0;
```

}

Zadatak: RGB



Zadatak

Petar je zaljubljenik u boje i šarenilo, a otkad je dobio računalo sve više ga zanima i računalna grafika. Danas se odlučio igrati s bojama zapisanim u RGB (Red Green Blue) formatu jer ga to jako zabavlja. Igra se tako da svakom RGB zapisu boje koji dobije želi izjednačiti vrijednosti R (crvene), G (zelene) i B (plave) nijanse, ali tako da doda ili oduzme što manje pojedine nijanse.

Ulazni podaci

U prvom retku se nalaze 3 cijela broja odvojena razmakom. Brojevi predstavljaju vrijednosti - **R, G i B** ($0 \leq R, G, B \leq 255$).

Izlazni podaci

U prvi i jedini redak treba ispisati jedan cijeli broj koji predstavlja ukupnu minimalnu "količinu" boje koju Petar mora dodati/oduzeti R, G i B vrijednostima tako da nakon dodavanja/oduzimanja one imaju iste vrijednosti.

Test primjer 1	Test primjer 2	Test primjer 3
Ulaz	Ulaz	Ulaz
123 125 123	121 129 122	7 173 212
Izlaz	Izlaz	Izlaz
2	8	205

Objašnjenje prvog test primjera: Mora drugoj nijansi G (zelenoj) smanjiti vrijednost za 2 da sve tri nijanse imaju istu vrijednost 123.

Objašnjenje drugog test primjera: prvoj nijansi mora povećati vrijednost za 1, a drugoj umanjiti za 7.

Vrijeme



Zadatak: Skokovi



Mirko se navukao na skijaške skokove i raznorazne teorije zavjere, te je počeo sumnjati u ispravnost računanja ocjena skijaških skokova. Zato moli vas da mu pomognete u tom zadatku.

Ukupna ocjena jednog skoka se sastoji od zbroja dviju komponenata.

- ☐ Pet sudaca ocjenjuje skok ocjenama za stil – svaka od pet ocjena je prirodni broj od 1 do 20. Odbacuje se najmanja i najveća ocjena, te se preostale tri zbroje i pridodaju ukupnoj ocjeni skijaša.
- ☐ Mjeri se duljina skoka u metrima. Na doletištu postoji kritična točka, a ovisno o veličini skakaonice ona je uvijek ili na 90 ili na 120 metara. Skijašu je cilj skočiti što dalje, a bodovi se računaju tako da skijaš dobije 60 bodova ako je skočio točno na kritičnu točku, za svaki metar više dobije 2 dodatna boda, a za svaki metar manje gubi 2 boda. Na primjer, ako je kritična točka na 120 metara i skijaš je skočio 93 metra, osvojiti će $60 - (27 \cdot 2) = 6$ bodova, a ako skoči 123 metra onda osvaja $60 + (3 \cdot 2) = 66$ bodova.

Napišite program koji, za zadane bodove sudaca i duljinu skoka, određuje koliko je ukupno bodova osvojio skijaš.

Vrijeme

ULAZNI PODACI:

- ☐ U prvom redu nalazi se prirodni broj K – pozicija kritične točke. Broj K je uvijek ili 90 ili 120.
- ☐ U drugom redu nalazi se prirodni broj D ($D \leq 200$) koji označava duljinu skoka u metrima.
- ☐ U trećem redu nalazi se točno pet prirodnih brojeva manjih ili jednakih od 20 koji predstavljaju bodove sudaca.



Zadatak: Skokovi



ULAZNI PODACI:

- U prvom redu nalazi se prirodni broj K – pozicija kritične točke. Broj K je uvijek ili 90 ili 120.
- U drugom redu nalazi se prirodni broj D ($D \leq 200$) koji označava duljinu skoka u metrima.
- U trećem redu nalazi se točno pet prirodnih brojeva manjih ili jednakih od 20 koji predstavljaju bodove sudaca.

IZLAZNI PODACI:

- U prvi i jedini red potrebno je ispisati broj ukupan bodova koji je skakač osvojio.
- Napomena: Primijetite da ukupan broj bodova može biti negativan.

ulaz

120

130

4 15 11 20 10

izlaz

116

ulaz

120

111

15 15 10 10 20

izlaz

82

ulaz

90

10

1 2 3 4 5

izlaz

-91

Vrijeme



Sortiranje



- ❑ Redanje elemenata u nekom redosljedu
- ❑ Razni algoritmi koji se razlikuju po brzini i složenosti izvedbe
 - sortiranje biranjem (*selection sort*)
 - *bubble sort*
 - sortiranje umetanjem (*insertion sort*)
 - *Shellov sort*
 - *mergesort*
 - *quick sort*
 - sortiranje s pomoću gomile (*heap sort*)

Sortiranje biranjem



- pronadi najmanji element niza i zamijeni ga s prvim elementom niza
- ponavljaj s ostatkom niza, smanjujući nesortirani dio

6	4	1	8	7	5	3	2
1	4	6	8	7	5	3	2
1	2	6	8	7	5	3	4
1	2	3	8	7	5	6	4
1	2	3	4	7	5	6	8
1	2	3	4	5	7	6	8
1	2	3	4	5	6	7	8
1	2	3	4	5	6	7	8
1	2	3	4	5	6	7	8

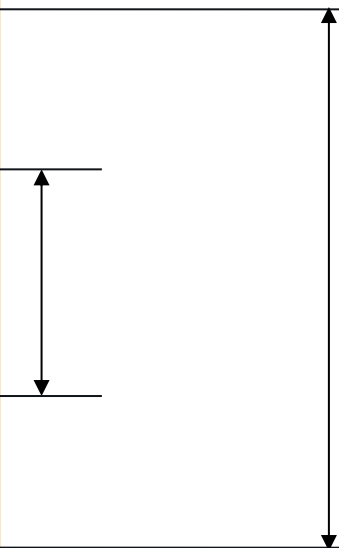
Sortiranje biranjem



□ izvedba - 2 petlje:

- vanjska određuje granice sortiranog dijela niza
- unutarnja traži najmanji element u nizu

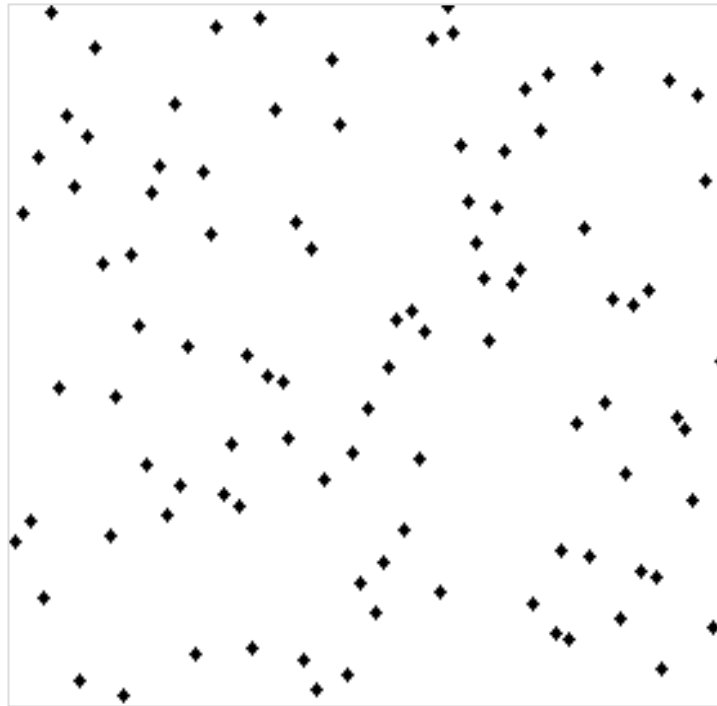
```
void SelectionSort (int A [], int N) {  
    int i, j, min;  
    for (i = 0; i < N; i++) {  
        min = i;  
        for (j = i+1; j < N; j++) {  
            if (A[j] < A[min]) min = j;  
        }  
        // Zamijeni A[i] i A[min]  
    }  
}
```



Bubble sort



- ❑ ideja: zamjena susjednih elemenata ako nisu u dobrom redoslijedu
 - kreni od početka niza prema kraju
 - zamijeni 2 elementa ako je prvi veći od drugog



Bubble sort



1. prolaz

6 4 1 8 7 5 3 2
4 6 1 8 7 5 3 2
4 1 6 8 7 5 3 2
4 1 6 8 7 5 3 2
4 1 6 7 8 5 3 2
4 1 6 7 5 8 3 2
4 1 6 7 5 3 8 2
4 1 6 7 5 3 2 8

2. prolaz

4 1 6 7 5 3 2 8
1 4 6 7 5 3 2 8
1 4 6 7 5 3 2 8
1 4 6 7 5 3 2 8
1 4 6 5 7 3 2 8
1 4 6 5 3 7 2 8
1 4 6 5 3 2 7 8

3. prolaz

1 4 6 5 3 2 7 8
1 4 6 5 3 2 7 8
1 4 6 5 3 2 7 8
1 4 5 6 3 2 7 8
1 4 5 3 6 2 7 8
1 4 5 3 2 6 7 8

4. prolaz

1 4 5 3 2 6 7 8
1 4 5 3 2 6 7 8
1 4 5 3 2 6 7 8
1 4 3 5 2 6 7 8
1 4 3 2 5 6 7 8

5. prolaz

1 4 3 2 5 6 7 8
1 4 3 2 5 6 7 8
1 3 4 2 5 6 7 8
1 3 2 4 5 6 7 8

6. prolaz

1 3 2 4 5 6 7 8
1 3 2 4 5 6 7 8
1 2 3 4 5 6 7 8

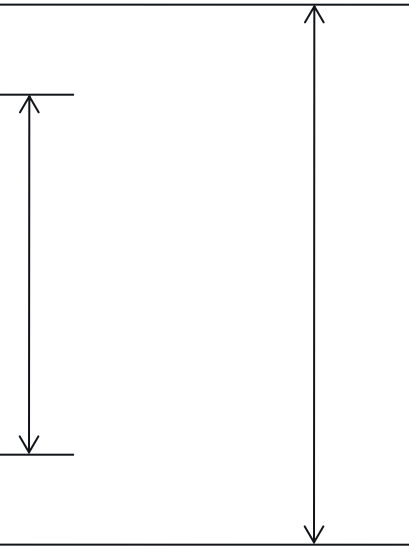
7. prolaz

1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8

Bubble sort



```
void BubbleSort (int A [], int N) {  
    int i, j;  
    for (i = 0; i < N-1; i++) {  
        for (j = 0; j < N-1-i; j++) {  
            if (A[j+1] < A[j])  
                // Zamijeni A[j] i A[j+1]  
        }  
    }  
}
```

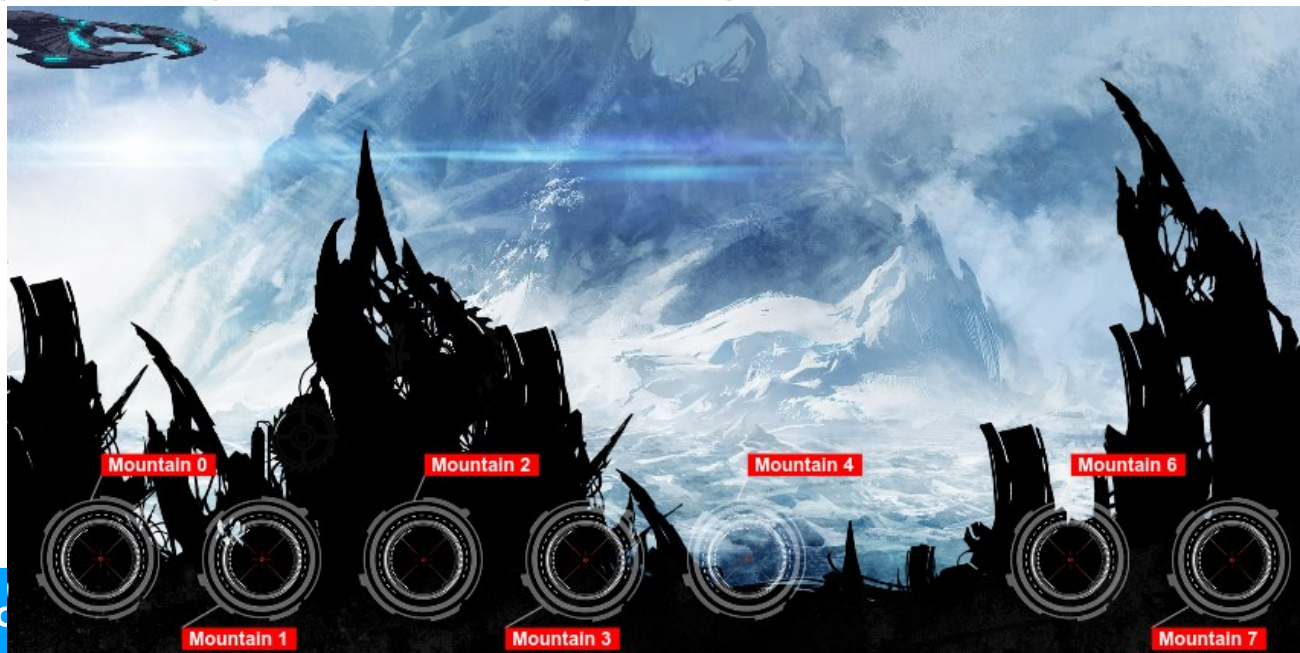


Kirk's Quest - The Descent



<http://www.codingame.com/ide/765073a27b1daf4fdd5f3b2ad2baedd36b80b3>

- ❑ Vaš zadatak je isprogramirati topove da unište brda prije nego brod u njih.
- ❑ Ukupno je 8 brda i brod kruži iznad njih s lijeva u desno, pa s desna u lijevo.
- ❑ Tijekom svakog kruga, brod se spušta 1 km, a igra ima ukupno 8 krugova. Tijekom svakog kruga možete pucati samo jednom na brdo koje se nalazi direktno ispod broda.



Kirk's Quest - The Descent



ULAZNI PODACI:

- ☐ Dva cijela broja SX i SY . $0 \leq \mathbf{SX} \leq 7$, $0 < \mathbf{SY} \leq 10$
 - SX predstavlja horizontalnu poziciju broda, a SY trenutnu visinu na kojoj se brod nalazi
- ☐ Sljedećih 8 linija sadrže po 1 cijeli broj MH koji predstavlja visinu pojedinog brda $0 \leq \mathbf{MH} \leq 9$

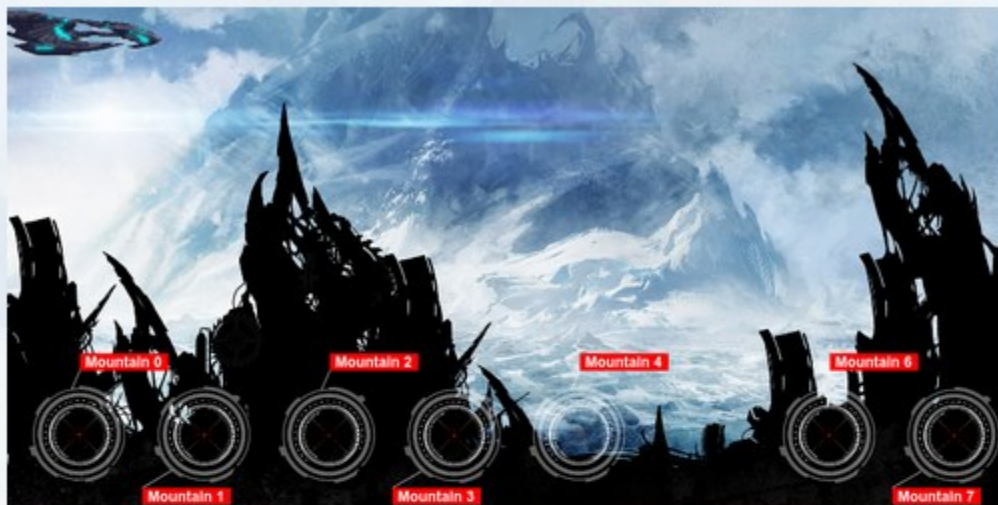
IZLAZNI PODACI svakog kruga:

- ☐ Jedna linija koja sadrži tekst **FIRE** (ako brod puca) ili **HOLD** (brod ne puca)

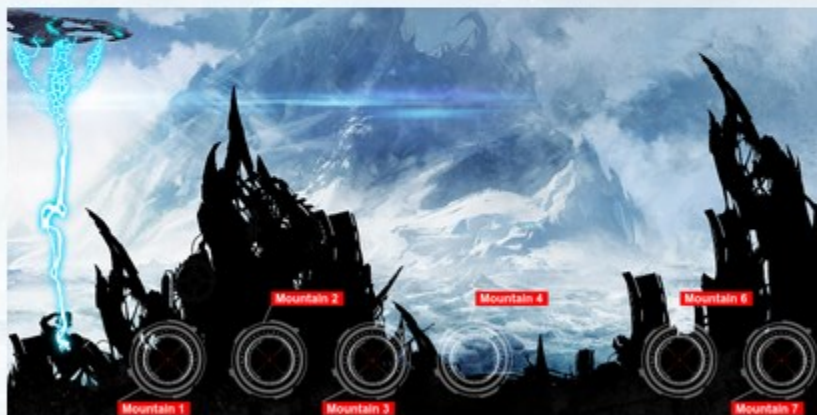
Vrijeme



Kirk's Quest - The Descent



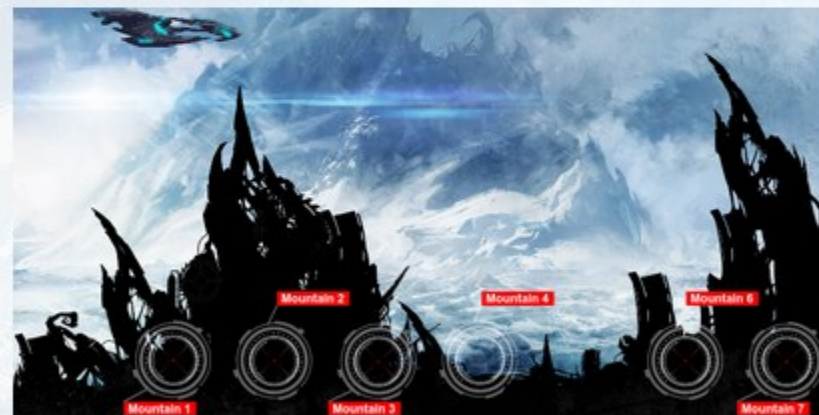
EXAMPLE: SHIP STARTS ON THE LEFT ($SX = 0$)
AT THE HIGHEST ALTITUDE ($SY = 10$)



TURN 1

FIRST MOUNTAIN ON THE LEFT IS 5 KILOMETERS HIGH.

FIRE STARSHIP FIRES.



TURN 2

SHIP HAS MOVED ONE POSITION TO THE RIGHT ($SX = 1$).

HOLD STARSHIP IS NOT FIRING