

UNIVERZITET U BEOGRADU

# ELEKTROTEHNIČKI FAKULTET

*Katedra za elektroniku*

*Predmet: Računarska elektronika*



## Projekat: 18 Moj Broj

Projekat radili:

Ime	Prezime	broj indeksa
Uroš	Cvjetinović	2016/0093
Nikola	Jugović	2016/0408

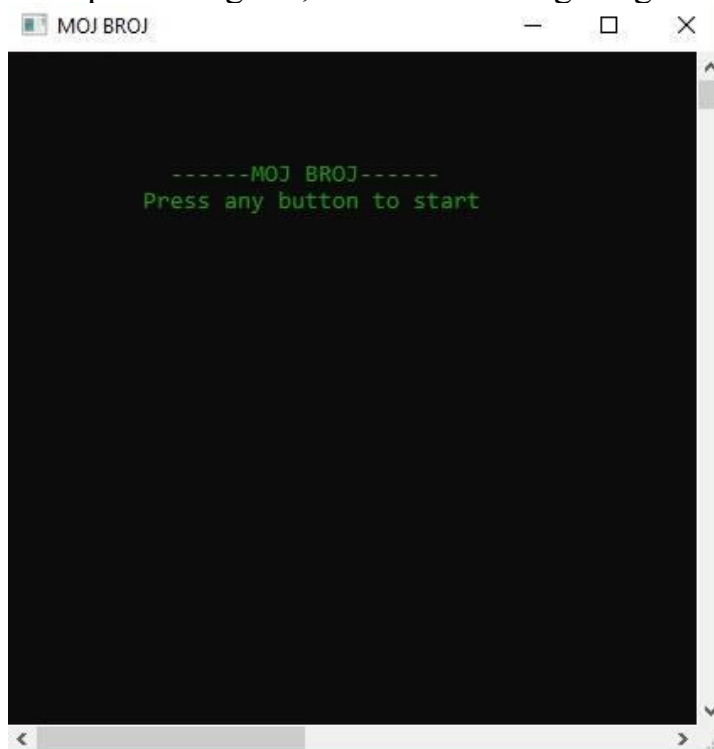
### Teskt zadatka:

Potrebno je realizovati igru “Moj broj” iz kviza “Slagalice”. Kada se započne igra, generiše se nasumičan broj u opsegu 1-999, kao i četiri broja u opsegu 1-9, jedan broj iz skupa {10, 15, 20} i jedan broj iz skupa {25, 50, 75, 100}. Potrebno je dobiti prvi generisani broj koristeći se isključivo ostalim generisanim brojevima (svaki broj se sme koristiti samo jednom), osnovnim računskim operacijama (sabiranje, oduzimanje, množenje i deljenje) i zagradama. Potrebno je sve generisane brojeve (osim prvog) ispisati u zasebnim poljima i omogućiti kretanje kursora po njima i njihov izbor pritiskom na odgovarajuće tastere. Nakon što se broj izabere, onemogućiti novi izbor istog broja. Obezbediti zasebna polja i za svaku od operacija i zgrade. Pritiskom na taster „space”, bira se broj/operacija na kojoj se kursor nalazi. Jasno označiti polje na kojem se nalazi kursor. Potrebno je i ispisati matematičku formulu koju igrač koristi za dobijanje traženog broja. Nakon što igrač završi sa ispisivanjem formule, pritiskom na taster „enter” rezultat se izračunava i igraču se šalje poruka ukoliko je uspeo da pogodi tačan broj ili koliko mu fali do tačnog broja.

### Realizacija aplikacije:

Program je napisan koristeći Irvine32.inc biblioteku i Irvinove preporuke za modele koji se koriste i veličinu steka za rad u VS2017.

Pri pokretanju aplikacije otvara se „Menu“ prozor, koji pritiskom na bilo koje dugme osim ESC pokreće igricu, dok taster ESC gasi igricu.



Do prozora u kom se igra „Moj broj“ se dolazi iz prozora „Menu“.



Gornji broj je traženi broj, dok brojevi ispod su brojevi koji su na raspolaganju. U toku igre ako se iskoristi neki broj on više nije dostupan i menja se boja teksta kojom je ispisan iz zelene (na raspolaganju) u crvenu (iskorišćen broj). Belo polje ispod nasumično generisanih brojeva je traka u koju korisnik upisuje izraz. Ispod polja za izraz nalaze se operacije, zagrade, koje u zavisnosti od toga da li trenutno mogu da se iskoriste u izrazu menjaju boju: crna boja (na raspolaganju), siva boja (zabranjen unos). Ovim ograničavanjem korisnika se izbegava neispravan unos formule. Korisnik upisuje formulu korišćenjem tastera „space“ i strelica. Strelicama se kreće po poljima koja može uneti u formulu pritiskom na taster „space“. Dodata su dva polja, koja su van opsega zadatka, polje „DEL“, koje briše prethodni uneti simbol/broj u formulu, i polje „CE“ koja pokreće igru ispočetka sa istim generisanim brojevima. Ovim je olakšan unos korisniku.



Pritiskom na taster „enter“ se objavljuje rezultat unetog izraza, i apsolutna vrednost razlike rezultata i traženog broja.



## Realizacija koda:

Kod je realizovan modularno. Iz više nezavisnih procedura je realizovan interfejs korisnika i igre i izračunavanje krajnjeg rezultata. Pomoću globalnih promenljivi Selected, Used, Changed, CursorPos, Numbers, Formula, FormulaSize, NeedToClose

Inicijalno generisanje nasumičnih brojeva je realizovano procedurama u **\_InitLogic.asm** (*MyRandom, InitLogic, RestartVar*).

- MyRandom(dword) – prosleđuje joj se 32bitni broj koji određuje maksimalnu vrednost nasumičnog broja. Povratna vrednost je broj koji može da uzme vrednosti između 1 i prosleđenog broja
- InitLogic(PTR WORD) – prosleđuje se pokazivač na početak niza u koji se smeštaju nasumično generisani brojevi koji odgovaraju zadatim opsezima

Pomoćne procedure koje su omogućile efikasnije iscrtavanje i pregledniji kod **\_Verifier.asm** (*DecToBin, CheckIfUsed, CheckIfSelected, CheckIfChanged*)

- DecToBin(BYTE) – prosleđuje se decimalni broj koji se konvertuje u broj  $2^{(N-1)}$ , gde je N prosleđeni broj. Pomoćna funkcija koja olakšava proveru zabranjenosti nekog polja
- CheckIfUsed(BYTE) – proverava da li je polje koje se razmatra (prosleđeni broj) već iskorišćen
- CheckIfSelected(BYTE) – proverava da li je selektovano polje koje je prosleđeno
- CheckIfChanged(WORD) – proverava da li treba polje koje je prosleđeno da se iscrta ponovo

Procedure pomoću kojih se iscrtavaju polja pri igranju

**\_Output.asm** (*CrateBlock, CrateDecLabeledBlock, CrateCharLabeledBlock, CrateStrLabeledBlock*)

- CrateBlock(BYTE, BYTE, BYTE, BYTE, BYTE) – procedura koja iscrta obojeni pravougaonik sa parametrima redom prosleđenim proceduri: gornja leva ivica (x,y), dužina stranica (x,y) i boja (1-crvena, 2-zelena, 3 – siva, 4 – crna, 5 – plava, 6 – bela). Pravougaonik se iscrta tako što se prazni znakovi ' ' sa određenom obojenom pozadinom ispisuju od početne ivice udesno do određene dužine, nakon čega se vrati do leve stranice bez ispisivanja znakova, spusti se za jedan simbol niže i iscrta ponovo nadesno, i tako sve dok ne stigne do određene dužine vertikalne ivice.
- CrateDecLabeledBlock (BYTE, BYTE, BYTE, BYTE, BYTE, BYTE, WORD, BYTE, BYTE) – Uže povezana procedure za funkcionalnost igre.

Kreira se pravougaonik sa brojem unutar njega pozivom `CrateBlock` koji poziciju i veličinu dobija direktno od prosleđenih argumenata, dok boju bloka i broja koji će se upisati u njega određuju poslednja dva argumenta. Prva četiri argumenta su pozicija i veličina potrebna za iscrtavanje bloka, druga dva određuju poziciju broja. Sedmi argument je broj koji se upisuje u pravougaonik. Poslednja dva argumenta predstavljaju redom, da li je polje selektovano (prosleđena 1) i da li je polje iskorišćeno. Zavisno od prosleđene vrednosti, ako je polje selektovano polje je obojeno plavom bojom, ako nije onda je sivom. Ako je polje iskorišćeno (prosleđena 1) broj koji se upisuje se upisuje crvenom bojom, dok ako nije onda se upisuje zelenom.

- `CrateCharLabeledBlock (BYTE, BYTE, BYTE, BYTE, BYTE, BYTE, BYTE, BYTE, BYTE)` – Procedura je slična proceduri `CrateDecBlockLabeled` u postupku iscrtavanja i prosleđivanja argumenata. Razlikuje je što se upisuje karakter (namenjeno za operacije „+,-,\*,/,(),“ ) a ne broj unutar pravougaonika i što se boja da je iskorišćeno polje siva, dok ako je raspoloživo onda je crno
- `CrateStrLabeledBlock (BYTE, BYTE, BYTE, BYTE, BYTE, BYTE, PTR BYTE, BYTE, BYTE)` – Procedura je slična proceduri `CrateCharLabeledBlock` u postupku iscrtavanja i prosleđivanja argumenata. Razlikuje se u tome što ne ispisuje karakter već prosleđen string. (Namenjeno za „DEL“ i „CE“)
- `Refresh (PTR WORD)` – Prosleđuje joj se pokazivač na početak niza u kojem su smešteni generisani brojevi. Glavna procedura za iscrtavanje layouta igre, poziva se kada treba da se osveži interfejs, što se dešava svakim pritiskom tastera. Procedura proverava da li polje treba da se iscrta ponovo (da li je promenjeno), što se dešava ako je bilo selektovano pa se kursor pomerio sa njega ili u slučaju da se tek selektovao, ili ako se izabrao pritiskom na taster „space“ ili ako je potrebno zabraniti operaciju nakon što se druga upravo izabrala. Ako polje treba da se promeni, proveriti se na koji način treba i pozove se njoj odgovarajuća procedura za iscrtavanje. U ovoj proceduri se i ispisuje izraz koji korisnik sastavlja, tako što globalnom `CursorPos` se prati položaj u izrazu, i zavisno od unetog simbola adekvatno upisuje ili briše unutar tog polja, tako što se dopisuje u izraz i povećava `CursorPos` ili preboji belom bojom i smanji `CursorPos`. Nakon brisanja se adekvatno nameštaju zabrane i iskorišćenosti svih polja.

Procedure koje su najuže povezane sa igrom Moj broj

**\_GameMechanic.asm**(*Menu, StartGame, PlayGame, Results*)

- Menu – Procedura setuje veličinu i naslov prozora i predstavlja početnu stranu za korisnika. Pritiskom na dugme koje nije ESC odvodi korisnika na stranu pri kojoj može započeti igru.
- StartGame – Pozivaju se *RestartVar*, *InitLogic*, *Refresh* kojim se podešavaju promenljive za početak igre.
- PlayGame – U ovoj proceduri se prati unos korisnika, i u zavisnosti od prethodnih i trenutnog stanja adekvatno namešta da li su polja zabranjena, iskorišćena i promenjena, nakon čega poziva proceduru *Refresh* koja iscrtava polja koja su promenjena ponovo. Pamti trenutno stanje izraza u globalnu promenljivu *Formula* (niz) u kojoj su smešteni redni brojevi polja koja su aktivirana onim redom kojim su se aktivirali. U slučaju da korisnik pokušava zabranjeno polje da izabere, program će ignorisati taj pokušaj i neće se iscrtavati iznova. Kraj unosa je kad se pritisne taster „enter“
- Results – Izračunava izraz koji je korisnik sastavio, koristi prostiji algoritam za izračunavanje za bolje performanse zbog kojeg program zahteva korišćenje zagrada. Izraz se nalazi u globalnoj promenljivoj *Formula*, i u njoj su smešteni redni brojevi polja, čime se lakše određuje prioriternost pri računanju, (broj 1 je namenjen za vrednost koja je izračunata i na steku se nalazi). Zbog svega mogućih sedam brojeva za kombinovanje koristi se sledeći algoritam za izračunavanje izraza: pronade se prva najprioritetnija zagrada na koju se naiđe (krećući se ->); unutar te zagrade traži prvo množenje/deljenje na koje se naiđe (krećući se ->); izračuna se vrednost i stavi na stek (ako operand bio na steku, skini sa steka i računaj sa njim); ponovi se pretraga i računanje množenja/deljenja; ako nema više množenja/deljenja ponovi se postupak za sabiranje/oduzimanje isto kao i za množenje/deljenje samo što se sad kreće ( <-); izračuna se vrednost i stavi na stek (ako operand bio na steku, skini sa steka i računaj sa njim); ponovi se pretraga i računanje sabiranje/oduzimanje svaki put kad se izračuna podizraz prepisu se 1 na mesta operanada I operacije; prepisi 1 preko zagrade čiji se izraz izračunao; ponovi postupak za sledeće zagrade; ako ne postoje više zagrade, prođe kroz računanje još jednom kao da su zagrade na početku I na kraju izraza. Izračuna vrednost I ispiše na ekran korisniku, I razliku između traženog I izračunatog broja. Ukoliko je izračunat broj negativan, ispisuje se 0
- RestartVar – Resetuje globalne promenljive za ponovni početak igre.