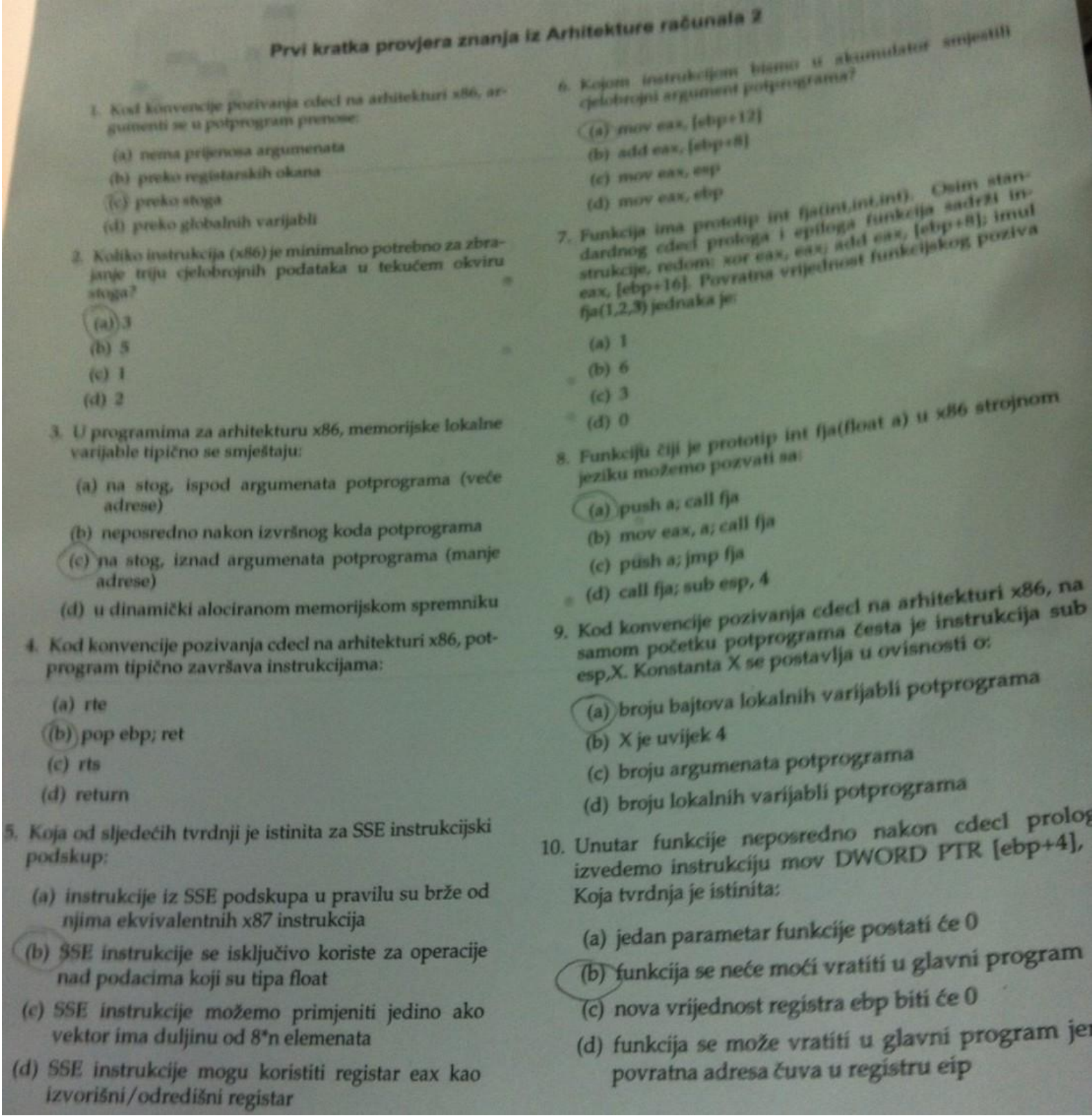
Prvi kratka provjera znanja iz Arhitekture računala 2

1. Kod konvencije pozivanja cdecl na arhitekturi x86, na samom početku potprograma česta je instrukcija sub esp,X. Konstanta X se postavlja u ovisnosti o:
   (a) X je uvijek 4
   (b) broju argumenata potprograma
   (c) broju bajtova argumenata potprograma
   (d) broju bajtova lokalnih varijabli potprograma
2. Koja od sljedećih tvrdnji je istinita za SSE instrukcijski podskup:
   (a) SSE instrukcije se isključivo koriste za operacije nad podacima koji su tipa float
   (b) SSE instrukcije mogu koristiti registar eax kao izvorišni/odredišni registar
   (c) instrukcije iz SSE podskupa u pravilu su brže od njima ekvivalentnih x87 instrukcija
   (d) SSE instrukcije možemo primjeniti jedino ako vektor ima duljinu od 8*n elemenata
3. Potprogram P koristi 1 lokalnu varijablu tipa char*. Za koliko je minimalno B potrebno pomaknuti esp u prologu potprograma P?
   (a) 7
   (b) 1
   (c) 12
   (d) 8
4. Kako bi se omogućili pozivi s varijabilnim brojem argumenata (npr. printf) konvencija pozivanja cdecl podrazumijeva da stogovni okvir s argumentima čisti:
   (a) pozvani kôd (potprogram), neposredno nakon pozivanja
   (b) potprogram ili glavni program, svejedno
   (c) pozivatelj (glavni program)
   (d) namjenski potpprogram (garbage collector)
5. Kojim odsječkom strojnog jezika x86 bismo izračunali a*b+c, ako su a, b i c smješteni na stogu?
   (a) add eax,[ebp+8]; imul eax,[ebp+16]; mov eax,[ebp+12];
   (b) mov eax,[ebp+12]; add eax,[ebp+8]; imul eax,[ebp+16]
   (c) imul eax,[ebp+16]; mov eax,[ebp+12]; add eax,[ebp+8]
   (d) mov eax,[ebp+8]; imul eax,[ebp+12]; add eax,[ebp+16]
6. Funkciju čiji je prototip int fja(float a) u x86 strojnom jeziku možemo pozvati sa:
   (a) push a; call fja
   (b) call fja; sub esp, 4
   (c) mov eax, a; call fja
   (d) push a; jmp fja
7. U programima za arhitekturu x86, memorijske lokalne varijable tipično se smještaju:
   (a) na stog, iznad argumenata potprograma (manje adrese)
   (b) neposredno nakon izvršnog koda potprograma
   (c) u statički alociranom memorijskom spremniku
   (d) na stog, ispod argumenata potprograma (veće adrese)
8. Unutar funkcije neposredno nakon cdecl prologa izvedemo instrukciju mov DWORD PTR [ebp+4], 0. Koja tvrdnja je istinita:
   (a) nova vrijednost registra ebp biti će 0
   (b) jedan parametar funkcije postati će 0
   (c) funkcija se neće moći vratiti u glavni program
   (d) nova vrijednost registra esp biti će 0
9. Kojom instrukcijom bismo u akumulator smjestili cjelobrojni argument potprograma?
   (a) mov eax, [ebp+12]
   (b) imul eax, [ebp+16]
   (c) mov eax, esp
   (d) add eax, [ebp+8]
10. Kod konvencije pozivanja cdecl na arhitekturi x86, potprogram tipično počinje instrukcijama:
    (a) push esp; mov esp, ebp
    (b) push esp; mov ebp, esp
    (c) push ebp; mov ebp, esp
    (d) push ebp; mov esp, ebp

1 1
3 12
6 16



Prvi kratka provjera znanja iz Arhitekture računala 2

1. Kod konvencije pozivanja cdecl na arhitekturi x86, argumenti se u potprogram prenose:
   (a) nema prijenosa argumenata
   (b) preko registarskih okana
   (c) preko stoga
   (d) preko globalnih varijabli

2. Koliko instrukcija (x86) je minimalno potrebno za zbrajanje triju cjelobrojnih podataka u tekućem okviru stoga?
   (a) 3
   (b) 5
   (c) 1
   (d) 2

3. U programima za arhitekturu x86, memorijske lokalne varijable tipično se smještaju:
   (a) na stog, ispod argumenata potprograma (veće adrese)
   (b) neposredno nakon izvršnog koda potprograma
   (c) na stog, iznad argumenata potprograma (manje adrese)
   (d) u dinamički alociranom memorijskom spremniku

4. Kod konvencije pozivanja cdecl na arhitekturi x86, potprogram tipično završava instrukcijama:
   (a) rte
   (b) pop ebp; ret
   (c) rts
   (d) return

5. Koja od sljedećih tvrdnji je istinita za SSE instrukcijski podskup:
   (a) instrukcije iz SSE podskupa u pravilu su brže od njima ekvivalentnih x87 instrukcija
   (b) SSE instrukcije se isključivo koriste za operacije nad podacima koji su tipa float
   (c) SSE instrukcije možemo primjeniti jedino ako vektor ima duljinu od 8*n elemenata
   (d) SSE instrukcije mogu koristiti registar eax kao izvorišni/odredišni registar

6. Kojom instrukcijom bismo u akumulator smjestili cjelobrojni argument potprograma?
   (a) mov eax, [ebp+12]
   (b) add eax, [ebp+8]
   (c) mov eax, esp
   (d) mov eax, ebp

7. Funkcija ima prototip int fja(int,int,int). Osim standardnog cdecl prologa i epiloga funkcija sadrži instrukcije, redom: xor eax, eax; add eax, [ebp+8]; imul eax, [ebp+16]. Povratna vrijednost funkcijskog poziva fja(1,2,3) jednaka je:
   (a) 1
   (b) 6
   (c) 3
   (d) 0

8. Funkciju čiji je prototip int fja(float a) u x86 strojnom jeziku možemo pozvati sa:
   (a) push a; call fja
   (b) mov eax, a; call fja
   (c) push a; jmp fja
   (d) call fja; sub esp, 4

9. Kod konvencije pozivanja cdecl na arhitekturi x86, na samom početku potprograma česta je instrukcija sub esp,X. Konstanta X se postavlja u ovisnosti o:
   (a) broju bajtova lokalnih varijabli potprograma
   (b) X je uvijek 4
   (c) broju argumenata potprograma
   (d) broju lokalnih varijabli potprograma

10. Unutar funkcije neposredno nakon cdecl prolog izvedemo instrukciju mov DWORD PTR [ebp+4], Koja tvrdnja je istinita:
   (a) jedan parametar funkcije postati će 0
   (b) funkcija se neće moći vratiti u glavni program
   (c) nova vrijednost registra ebp biti će 0
   (d) funkcija se može vratiti u glavni program jer povratna adresa čuva u registru eip

1. SLIKA odgovori:
   1. D
   2. C
   3. D
   4. C
   5. D
   6. D
   7. A
   8. C
   9. A
   10. C
2. SLIKA – 7. C
   Neki odgovori zaokruženi na slici 2 nisu točni, ovo iznad je točno.