

UPOTREBA CHATGPT-JA ZA RJEŠAVANJE ISPITA IZ UVODA U PROGRAMIRANJE

USING CHATGPT TO SOLVE INTRODUCTION TO PROGRAMMING EXAM

Goran Đambić

Visoko učilište Algebra, Gradišćanska 24, 10000 Zagreb, Hrvatska

SAŽETAK

ChatGPT i ostali modeli umjetne inteligencije (UI) na velika su vrata ušli u mnoge dijelove života pa tako i u visokoškolsko obrazovanje. Osim prilika za učenje, modeli umjetne inteligencije studentima nude i mogućnosti za varanje na provjerama znanja te pred nastavnike postavljaju nove izazove. Mogućnosti ChatGPT-ja da generira programski kôd na temelju teksta zadatka na engleskome jeziku dobro su istražene. Radova koji se bave ponašanjem modela pri zadavanju zadataka na hrvatskome jeziku nema, stoga je cilj ovoga rada pobliže istražiti kakve su trenutne performanse modela ChatGPT 3.5 u tom području. Rad se sastoji od tri glavna dijela. U prvom dijelu se ChatGPT-ju zadaju zadaci s ispita iz kolegija Uvod u programiranje, u istom obliku u kakvom su bili zadani i studentima. U drugom dijelu analizira se koliko je uočljivo da su rješenja nastala automatskim generiranjem, a ne od strane studenta. Konačno, mijenjaju se upiti ChatGPT-ju i istražuje se je li moguće utjecati na izgled generiranih rješenja kako se ne bi razlikovala od studentskih rješenja.

Ključne riječi: *UI generirani programski kôd, ChatGPT upiti na hrvatskome jeziku, kolegij Uvod u programiranje, UI u visokome obrazovanju, disruptivnost u obrazovanju utjecajem UI*

ABSTRACT

ChatGPT and other artificial intelligence models have entered many aspects of life, including higher education. In addition to providing learning opportunities, artificial intelligence models offer

students the chance to cheat on tests, posing new challenges for teachers. The capabilities of ChatGPT to generate program code based on English input text are well-explored. However, there are no papers addressing the behavior of the model when tasked with writing programming assignments in Croatian. Therefore, the aim of this paper is to investigate more thoroughly the current performance of the ChatGPT 3.5 model in that area. The paper consists of three main parts. In the first part, exam tasks from the Introduction to Programming course are presented to the ChatGPT in the same format as they were given to students. In the second part, an analysis is conducted to assess how detectable it is that the solutions were created through automatic generation rather than by the student. Finally, the inputs to ChatGPT are modified, and an investigation is carried out to determine whether it is possible to influence the shape of the generated solutions so that they do not differ from student solutions.

Keywords: *AI generated program code, ChatGPT inputs in Croatian, course Introduction to programming, AI in higher education, disruption of education caused by artificial intelligence*

1. UVOD

1. INTRODUCTION

Prema riječima njega samoga, ChatGPT je veliki jezični model razvijen u kompaniji OpenAI i obučen na podacima javno dostupnim na internetu zaključno s rujnom 2021. godine. Temeljen je na neuronskoj mreži transformatorske arhitekture poznatoj kao GPT-3 koja sadrži velik broj ulaznih

parametara koji mu omogućuju razumijevanje upita i generiranje odgovora.

ChatGPT postavio je temelje sličnim tehnologijama koje su već počele uzrokovati poremećaje i izazvati drastične promjene u mnogim područjima, pa tako i u sustavu obrazovanja i znanosti [1]. Primjerice, cijeli prethodni odlomak gotovo je u potpunosti generirao ChatGPT. To nije nigdje naznačeno budući da ChatGPT ne može preuzeti odgovornost za generirani tekst te ga prema renomiranim izdavačima kao što su Science, Elsevier i Springer nije obavezno navoditi kao autora [2].

Pojava ChatGPT-ja uzrokovala je opravdanu zabrinutost za akademsku čestitost i prepisivanje u visokome obrazovanju [3]. Vrlo je lako zamisliti scenarij u kojemu student dobiva zadatak napisati esej. Umjesto da prouči preporučenu literaturu, student osmisli okvir te primijeni svoj znanje za izradu eseja jednostavnim kopiranjem teksta zadatka u ChatGPT. Kopiranjem generiranog odgovora, student za nekoliko sekundi ima gotov esej koji će zatim predati kao svoj rad. Sličan scenarij bio je ostvariv i prethodnih godina, ali je od studenta zahtijevao ozbiljniji angažman u pretraživanju interneta putem tražilica te kompiliranju radova iz raznih izvora. Zahvaljujući jednostavnom i javno dostupnome sučelju, danas je varanje dostupno svima u obliku postavljanja upita i klika na gumb. Kvalitativno istraživanje [4] pokazuje da bi više od 50% ispitanih studenata koristilo ChatGPT za generiranje sadržaja koje bi predali kao svoj rad. Uz prihvaćanje neosporne činjenice da se ChatGPT može koristiti za varanje, velik broj autora naglašava potrebu za redizajniranjem procesa podučavanja i vrednovanja kako bi se studentima omogućilo educirano korištenje ChatGPT-a kao vrijednog alata koji će im pomoći u svladavanju gradiva [5], [6] i [7].

Namjera je ovoga rada detaljnije proučiti potencijalno ozbiljne poremećaje uzrokovane upotrebom ChatGPT-ja u području vrednovanja osnova programiranja na razini 6 HKO. Ciljevi rada su provjeriti kako ChatGPT rješava trenutni ispit iz kolegija prijediplomske razine, konkretnije, kolegija Uvod u programiranje te je li očito da je ispit riješila umjetna inteligencija, a ne student početnik. Ako je ta činjenica očita, cilj

rada je istražiti je li moguće dodatnim uputama navesti ChatGPT da izradi rješenje koje se neće razlikovati od onoga kakvo bi napisao student početnik. Pri tome se zadaci zadaju na hrvatskome jeziku, kopiranjem izvornog teksta zadatka, bez izmjena.

Slični radovi postoje za zadatke na engleskome jeziku. Primjerice, autori u [8] zaključuju da ChatGPT uspješno polaže ispit iz kolegija Uvod u funkcijski programski jezik ocjenom 4, na 155. mjestu od ukupno 314 studenata. Autori u [9] zaključuju da je ChatGPT sposoban uspješno rješavati zadatke u programskome jeziku Java, uz očekivano ograničenje da formulacija zadataka mora biti tekstualna, bez slika ili grafova. Autori u [10] potvrđuju sposobnost ChatGPT-a da generira točna rješenja jednostavnijih problema u programskom jeziku Python, ali da mu performanse opadaju s porastom složenosti problema.

2. METODOLOGIJA

2. *METHODOLOGY*

S obzirom da je cilj ovoga rada proučiti mogućnosti koje studenti imaju za generiranje programskog kôda kao odgovara na ispitne zadatke, rad će biti ograničen na korištenje besplatne i svima dostupne verzije ChatGPT-3.5. Usluga je dostupna na <https://chat.openai.com/> te je odmah nakon registracije moguće postavljati upite. Sučelje za postavljanje upita vrlo je jednostavno, što sve zajedno čini ChatGPT univerzalno dostupnim i trivijalnim za korištenje. Primjerice, ako se postavi pitanje: „Koji je glavni grad Hrvatske?“, kroz koju sekundu, ovisno o opterećenju sustava, ChatGPT će generirati točan odgovor: „Glavni grad Hrvatske je Zagreb.“ Međutim, ChatGPT će iznenađujuće često dati krivi odgovor. Primjerice, postavljanjem pitanja: „Koji je hrvatski grad peti po broju stanovnika?“, ChatGPT će dati krivi odgovor: „Prema podacima iz mog znanja do rujna 2021. godine, peti najveći grad u Hrvatskoj po broju stanovnika bio je Osijek“. No, prema popisu stanovnika 2011. i 2021. peti grad u Hrvatskoj po broju stanovnika je Zadar, dok je Osijek četvrti.

Ovim radom želi se provjeriti na koji se način mogućnosti javno dostupnog ChatGPT-ja

moгу iskoristiti u akademskom kontekstu u polju računarstva, u kolegiju koji studentima daje osnovne vještine programiranja. Iako je provedeno mnogo sličnih istraživanja puno, prvenstveno su temeljena na postavljanju upita na engleskome jeziku. Ovaj će rad slijediti postupak zadavanja upita na hrvatskome jeziku, a odgovor se traži u obliku programskog kôda u programskom jeziku C++.

Ovaj rad temelji se na tri postavljene hipoteze:

- Hipoteza H1: ChatGPT sposoban je položiti ispit iz kolegija Uvod u programiranje ako mu su zadaci na hrvatskome jeziku.

Zadaci će biti upisani u izvornom obliku i bez ikakve izmjene, kako su zadani i studentima. ChatGPT će generirati rješenja u programskom jeziku C++ koja će biti ocijenjena po istim kriterijima kao i studentska rješenja.

- Hipoteza H2: Analizom strukture rješenja koja je generirao ChatGPT na temelju izvornih zadataka, postoje jasni pokazatelji da rješenje nije napisao student početnik koristeći jezične elemente koji su naučeni kroz predavanje, vježbe i obaveznu literaturu.
- Hipoteza H3: Dodatnim prilagodbama izvornih zadataka ChatGPT može generirati rješenja koja se ne mogu razlikovati od rješenja kakva bi napisao student početnik.

Hipoteza H1 ispitat će se na ispitu iz kolegija Uvod u programiranje iz 2022. godine. Zadaci će se pojedinačno kopirati u polje za unos teksta. Pri tome će svaki od zadataka nužno biti malo izmijenjen kako bi se definirao kontekst kojeg studenti podrazumijevaju na ispitu, ali ga ChatGPT-ju treba eksplicitno definirati. U svaki zadatak će se dodati prva rečenica koja glasi: „Napiši program u programskom jeziku C++ koji rješava sljedeći zadatak:“, iza čega slijedi izvorni tekst zadatka. Dobivena rješenja ocijenit će se prema istim kriterijima kao i rješenja studenata.

Kao podloga za hipotezu H2 iz sadržaja predavanja, vježbi i obavezne literature bit će definirani elementi programskoga jezika za koje se očekuje da ih koriste prosječni studenti. Iz toga slijedi da svi elementi programskog jezika koji ne pripadaju tom skupu spadaju u kategoriju sumnjivih elemenata koji kod ocjenjivača trebaju izazvati sumnju da programski kôd nije pisao

student početnik. Primjerice, očekivano je da student varijablu deklarira eksplicitno navodeći tip podataka:

```
int a = 10;
```

Ako je u rješenju varijabla implicitno deklarirana:

```
auto a = 10;
```

to će biti indikacija da student možda nije samostalno napisao programski kôd. To ne znači da student sigurno nije samostalno napisao rješenje jer neki imaju više predznanja, uče iz dodatnih izvora, a možda i koriste ChatGPT za učenje. Međutim, indikacije mogu biti koristan alat za dodatno vrednovanje studentskog znanja usmenim odgovaranjem.

Hipoteza H3 bit će razmatrana tako što će se tekst ispitnih pitanja iterativno mijenjati dodavanjem i izmjenama teksta samog zadatka sve dok ChatGPT ne generira rješenje bez sumnjivih elemenata. Pretpostavka je da ako ChatGPT generira rješenje korištenjem samo očekivanih elemenata, takvo se rješenje neće moći razlikovati od rješenja studenta početnika. Pri tome treba uzeti u obzir da student koji na ovaj način modificira zadatke u traženju odgovora koji neće probuditi sumnju vjerojatno mora imati višu razinu znanja programiranja od početničke.

Ispit koji je korišten sastoji se od 17 zadataka podijeljenih u šest ishoda učenja i ukupno nosi 90 bodova. Prvih pet ishoda učenja sadrže po tri zadatka ukupne vrijednosti po 16 bodova, dok šesti ishod učenja sadrži dva zadatka ukupne vrijednosti 10 bodova. Smatra se da je student položio neki ishod učenja ako je ostvario barem 50% bodova na tom ishodu učenja, pri čemu se bodovi dodjeljuju djelomično za svaki zadatak prema unaprijed definiranoj shemi ocjenjivanja.

Kao prvi korak, svakom od zadataka dodan je tekst: „Napiši program u programskom jeziku C++ koji rješava sljedeći zadatak:“ te je dobiveni tekst zadatka kopiran u polje za unos teksta. Generirani rezultat zatim je kopiran u alat za razvoj računalnih programa u programskom jeziku C++. U alatu je program pokrenut te su dodijeljeni bodovi na temelju izvršavanja programa i njegove strukture. Primjerice, ako program dobro izračunava i ispisuje prosjek,

ali izračun ne radi u funkciji kako je traženo, dodjeljuju se djelomični bodovi.

Nakon dodjele bodova, svaki od generiranih programa analiziran je u potrazi za indikacijama da programski kôd nije pisao student početnik. Da bi to bilo moguće, definirana je baza podataka elemenata programskog jezika C++ koji su korišteni na predavanjima, vježbama i u obaveznoj literaturi: if, switch, for, while, bool, char, void, new, itd. Svi ostali elementi programskog jezika predstavljaju skup indikacija: static_cast, for-each petlja, const kod deklariranja varijabli i parametara, funkcija isdigit() i slično. Iako i navedeni indikativni elementi predstavljaju relativno osnovne jezične elemente, oni nisu dio kurikuluma i studenti ih nisu vidjeli na nastavi, stoga većina studenata nije upoznata s njima i ne zna ih koristiti na prvom programerskom kolegiju. Iz tog razloga njihovo se prisustvo tretira kao indikacija da programski kôd možda nije pisao student početnik.

Na kraju, kod svih zadataka čija rješenja imaju barem jednu indikaciju, pristupa se pokušaju stvaranja rješavanja koje neće sadržavati ikakvu indikaciju. Tekst zadatka ponovno se kopira u polje za unos teksta te se dodaje tekst koji od ChatGPT-ja traži da prilikom generiranja ili koristi određene elemente ili izbjegava određene elemente, s ciljem da novogenerirano rješenje bude bez indikacija. Za svaki zadatak pokušava se najviše 10 puta dobiti rješenje bez indikacija.

Tablica 1 Bodovi po ishodima učenja dodijeljeni ChatGPT rješenjima.

Table 1 Points per learning outcomes awarded to ChatGPT solutions.

Ishod učenja	Dobiveno bodova	Moguće bodova
Osmisliti jednostavan algoritam prema smjernicama te ga implementirati koristeći osnovne elemente programskog jezika	12	16
Odabrati prikladan kontejner za čuvanje više podataka te primijeniti složenije matematičke i logičke operacije na njegove elemente	13,5	16
Osmisliti rješenje jednostavnog problema koristeći funkcije te ih kreirati i upotrijebiti	12,5	16
Dizajnirati jednostavan korisnički tip podataka te koristiti njegove instance za rješavanje problema	15	16
Kreirati rješenje korištenjem raspoloživih memorijskih opcija stoga i hrpe	16	16
Konstruirati rješenje korištenjem datoteka	9	10
Ukupno:	78	90

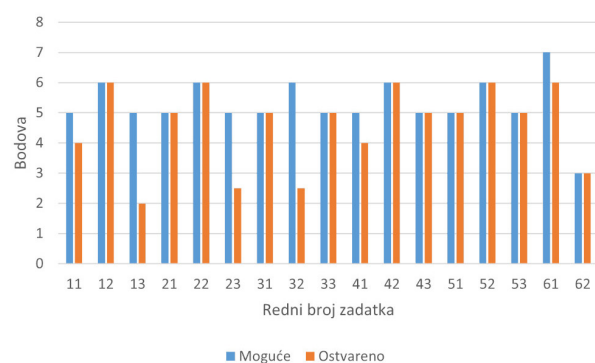
3. REZULTATI

3. RESULTS

Nakon ocjenjivanja svih zadataka, jasno je da je ChatGPT uspješno položio ispit sa 78 bodova od mogućih 90, odnosno, s dobivenih 87% bodova (Tablica 1), čime je potvrđena hipoteza H1.

Iz Slike 1 vidljivo je da je ChatGPT potpuno riješio 11 od 17 zadataka. Za tri zadatka generirao je rješenja koja su malo odstupala od traženih pa je stoga dobio po jedan bod manje od maksimuma, dok je na preostala tri zadatka napravio ozbiljnije propuste. Razlozi zbog kojih je ChatGPT gubio bodove su sljedeći:

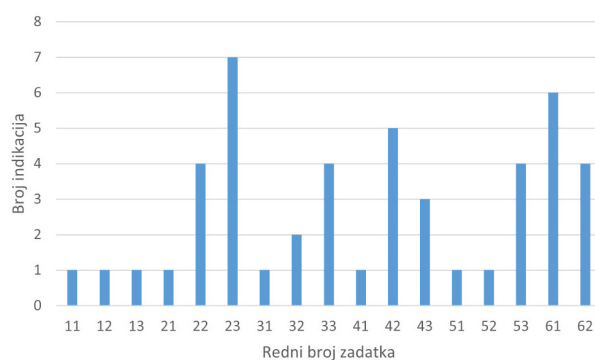
- U zadacima 11, 41 i 61 za učitavanje stringa s tipkovnice koristio je operator >> umjesto getline(), iako je iz teksta zadatka jasno da učitavani stringovi mogu sadržavati i razmake. Dodatno, u zadatku 11 ignorirao je zadani format ispisa na ekran.
- U zadatku 13 trebao je napisati programski kôd koji za učitani neparni cijeli broj A, u konzoli iscrtaava spojene znakove < i > visine A, ali generirani kôd iscrtaavao je nešto posve drugo.
- U zadatku 23 trebao je maskirano ispisati sve znamenke broja kreditne kartice osim zadnje četiri, ali u nekim slučajevima to nije ispravno funkcioniralo.
- U zadatku 32 trebao je pronaći najmanji zajednički višekratnik dvaju brojeva, ali za neke brojeve to nije ispravno funkcioniralo.



Slika 1 Dobiveni bodovi po ispitnim zadacima

Figure 1 Awarded points per exam tasks

Nakon zaključka da je ChatGPT uspješno položio ispit te potvrđivanja hipoteze H2, slijedi analiza broja indicacija da rješenja nije pisao student početnik već da su automatski generirana (Slika 2).



Slika 2 Broj indicacija po zadatku

Figure 2 Number of indications per task

Iz Slike 2 vidljivo je da u svakom generiranom rješenju postoji barem jedna indicacija. Razlog za to neuključivanje je imenskog prostora `std` jednom na početku programa pomoću „`using namespace std`“, već eksplicitno stavljanje prefiksa imenskog prostora pri svakoj upotrebi, primjerice „`std::cout`“, „`std::endl`“ i slično. S obzirom da skoro svi studenti početnici preferiraju „`using namespace std`“ i koriste ga za rješavanje zadataka na vježbama i školskim zadaćama, malo je vjerojatno da bi se na ispitu odlučili za drugu, dulju varijantu. Preostale indicacije pronađene u generiranim rješenjima su sljedeće:

- Parametar funkciji `srand()` drugačiji je od onoga s predavanja i vježbi.
- Učitavaju se podaci s tipkovnice, iako je iz teksta zadatka očito da treba podatke hardkodirati u rješenje, što je i brži način kojeg studenti uvijek preferiraju.

- Na neuobičajen se način koristi `cin.ignore()`.
- Koriste se funkcije `isdigit()`, `find()` i `reverse()` te metode `clear()` i `seekg()` na `ifstreamu`.
- Koristi se ternarni operator, operator+ za spajanje stringova te `sizeof`.
- Za pretvaranje tipova podataka koristi se `static_cast`.
- Koristi se `for-each` petlja.
- Koristi se `const` za deklariranje varijabli.
- Parametri se šalju po referenci i kao `const` kada to nije potrebno.
- U izradi polja objekata koriste se inicijalizacijske liste.
- Koristi se `tok cerr`.

Navedenim indicacijama potvrđena je hipoteza H2 jer analiza rješenja snažno sugerira da student početnik nije njihov autor.

Prvi korak u dokazivanju hipoteze H3 pokušaj je uklanjanja eksplicitnih prefiksa „`std::`“ sa svih elemenata u rješenju. To se pokazalo vrlo jednostavnim za postići uključivanjem rečenice „Uz korištenje `using namespace std;`“ i bez eksplicitnih prefiksa `std::` ispred svakog teksta zadatka. Tim jednostavnim potezom broj preostalih rješenja s indicacijama pao je na 9, a u nastavku će biti opisani pokušaji njihovog uklanjanja. Podebljano i ukošeno bit će prikazani novi elementi upita.

Izvorno rješenje zadatka 22 u sebi je uključivalo indicacije korištenja funkcije `srand()` na drugačiji način, korištenje `static_cast` pretvaranja tipova podataka te `for-each` petlje. Prvi pokušaj izbjegavanja indicacija glasio je: „Uz korištenje `using namespace std;`“ i bez eksplicitnih prefiksa `std::`, napiši program u programskom jeziku C++ koji rješava sljedeći zadatak: napišite program koji generira deset slučajnih brojeva između 1 i 100 i smješta ih u polje. Nakon toga, izračunajte prosjek tih brojeva pa prepisite sve brojeve veće od ili jednake prosjeku u vektor. Ispišite sve brojeve iz vektora. ***U rješenju nemoj koristiti `const`, pretvaranje tipova podataka niti `for-each` petlju. U pozivu funkcije `srand` koristi `time` i `nullptr`.*** Ovo je rezultiralo kôdom kojeg nije bilo moguće prevesti zbog deklariranja polja nekonstantnim izrazom, a uz to je ipak korišten `static_cast`, bez obzira na zahtjev. Opetovanim pokušajima uklanjanja svih indicacija, to nije bilo moguće ostvariti. Ili su sve indicacije uklonjene, ali je rezultat bio kôd koji se ne može prevesti,

ili se kôd može prevesti, ali sadrži indikacije za koje je eksplicitno traženo da ih nema. Nakon 10 pokušaja proglašen je neuspjeh i u najboljem slučaju je uvijek ostala po jedna indikacija.

Izvorno rješenje zadatka 23 u sebi je uključivalo nepotrebno učitavanje od korisnika, dvije for-each petlje, const i funkciju isdigit(). Logičnom modifikacijom zadatka riješene su sve indikacije: „... U rješenju nemojte koristiti funkciju isdigit, ključnu riječ const, for-each petlju.“

U zadatku 32 rješenje sadrži ternarni operator. Nikakvom formulacijom upita nije bilo moguće dobiti rješenje bez ternarnog operatora, stoga je nakon 10 pokušaja proglašen neuspjeh.

Zadatak 33 jednostavno je riješen dodavanjem teksta na kraj: „... U rješenju nemojte koristiti for-each petlju, niti const, niti prijenos parametara po referenci.“

Izvorno rješenje zadatka 42 sadržavalo je indikacije kod izrade polja inicijalizacijskim listama i bez eksplicitnog navođenja veličine te korištenje const i sizeof. Modificiranjem upita tako da glasi „... U rješenju nemojte koristiti const, sizeof niti for-each petlju. Kod izrade polja objekata nemojte koristiti inicijalizacijske liste već naknadno eksplicitno dodijelite vrijednosti svakom članu.“ postignut je značajan uspjeh, ali se niti nakon deset pokušaja nije uspjela ukloniti indikacija const.

Jednostavnim modificiranjem upita u zadatku 43 jednakim kao u prethodnim zadacima uklonjene su sve indikacije.

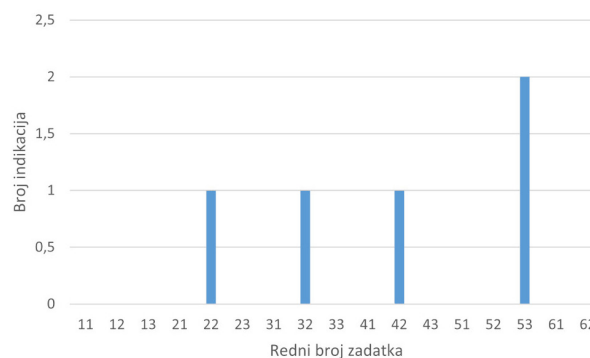
Rješenje zadatka 53 sadržavalo je sizeof, const i funkciju find(). Nikakva promjena upita nije uspjela ukloniti sizeof, dok su neke varijacije uklonile preostale tražene elemente, ali ne sve. Čak su i uvele for-each petlju, uvijek rezultirajući s barem dvije indikacije.

Svih šest indikacija zadatka 61 jednostavno je uklonjeno zadavanjem „... Nemojte koristiti cerr već grešku ispišite u cout. Nemojte koristiti reverse() već čitajte od kraja prema početku. Nemojte koristiti for-each petlju niti const.“

Sve indikacije zadatka 62 uklonjene zadavanjem su „... Nemojte koristiti cerr već grešku ispišite u cout. Koristite vektor za čuvanje objekata. Svaki broj u datoteci pročitajte samo jednom. Nemojte koristiti

for-each petlju niti funkciju za izračun površine.“

Slika 3 prikazuje količinu preostalih indikacija nakon napravljenih izmjena u tekstu zadataka. Iz slike je vidljivo da bez obzira na napravljene promjene, u rješenjima ostaje barem nekoliko indikacija, čime je hipoteza H3 opovrgnuta.



Slika 3 Količina preostalih indikacija

Figure 3 Number of remaining indications

4. ZAKLJUČAK

4. CONCLUSION

U radu je pokazano da je besplatna i široko dostupna inačica ChatGPT-ja više nego sposobna riješiti zadatke kolegija Uvod u programiranje zadane na hrvatskom jeziku te položiti kolegij velikim brojem bodova.

Kopiranje zadataka u polje za unos teksta i zatim kopiranjem rješenja natrag u alat za programiranje rezultira kôdom za koji postoje očite indikacije da ga nije pisao student početnik. Prije svega, to se odnosi na učestalo korištenje prefiksa „std::“, što je uobičajena praksa profesionalaca, ali nipošto nije uobičajena praksa studenata kolegija. Također, korištenje ključnih riječi const, static_cast, funkcija iz zaglavlja algorithm i slično, ChatGPT generira kôd koji je u skladu s pravilima struke, ali je izvan dometa programera početnika na kolegiju. Stoga je ovakvo korištenje ChatGPT-ja jednostavno za prepoznati i može poslužiti kao temelj za pozivanje studenata na dodatnu usmenu verifikaciju ispita.

Treći dio rada bavio se mogućnostima da se malenim promjenama teksta zadatka uklone indikacije iz rješenja. Pokazalo se da takav pristup ima dva rezultata – neke indikacije vrlo su se jednostavno uklonile dodavanjem teksta što želimo („koristite vektor za čuvanje objekata“) ili ne

želimo („nemojte koristiti cerr već grešku ispišite u cout“) dobiti u rješenju, dok se neke indikacije nikakvim parafraziranjem teksta nisu mogle ukloniti (najočitiji primjer definiranje je veličine polja gdje je rezultat bio definiranje const varijablom ili bez nje, ali s kôdom koji se nije dao prevesti).

Stoga je zaključak da pažljiviji student može jednostavno ukloniti veliku većinu indikacija, ali ne sve te se preporuča ispravljajima obratiti pažnju na takve situacije, naravno u kontekstu njihovog kolegija. Dodatno, uvođenjem slika ili tablica u tekst zadatka studentima se neće promijeniti težina ispita, ali će se time onemogućiti generiranje rješenja od strane besplatne inačice ChatGPT-a.

5. REFERENCE

5. REFERENCES

- [1.] García-Peñalvo F. J.; 2023; The perception of Artificial Intelligence in educational contexts after the launch of ChatGPT: Disruption or Panic?; *Education in the Knowledge Society*, 24, Article e31279. ISSN: 2444-8729. <https://doi.org/10.14201/eks.31279>
- [2.] Ekanayake K.B.; 2023; ChatGPT: To cite or not to cite?; *Sri Lanka Journal of Forensic Medicine, Science & Law*, 14(1), p.1-2. DOI: <https://doi.org/10.4038/sljfmsl.v14i1.7942>
- [3.] Sullivan M.; Kelly A.; McLaughlan P.; 2023; ChatGPT in higher education: Considerations for academic integrity and student learning. 6. 10.37074/jalt.2023.6.1.17.
- [4.] Ventayen R. J. M.; 2023; ChatGPT by OpenAI: Students' Viewpoint on Cheating using Artificial Intelligence-Based Application (February 16, 2023); Available at SSRN: <https://ssrn.com/abstract=4361548> or <http://dx.doi.org/10.2139/ssrn.4361548>
- [5.] Cotton D. R. E.; Cotton P. A.; Shipway J. R.; 2023; Chatting and cheating: Ensuring academic integrity in the era of ChatGPT; *Innovations in Education and Teaching International*, DOI: 10.1080/14703297.2023.2190148
- [6.] Anders B.A.; 2023; Is using ChatGPT cheating, plagiarism, both, neither, or forward thinking?; *Patterns* (N Y). 2023 Feb 28;4(3):100694. doi: 10.1016/j.patter.2023.100694. PMID: 36960444; PMCID: PMC10028419.
- [7.] Benuyenah, V.; 2023; Commentary: ChatGPT use in higher education assessment: Prospects and epistemic threats; *Journal of Research in Innovative Teaching & Learning*, Vol. 16 No. 1, pp. 134-135. <https://doi.org/10.1108/JRIT-03-2023-097>
- [8.] Geng, C.; Yihan, Z.; Pientka, B.; Si, X.; 2023; Can ChatGPT Pass An Introductory Level Functional Language Programming Course?; arXiv:2305.02230, DOI: <https://doi.org/10.48550/arXiv.2305.02230>
- [9.] Ouh E.L.; Gan B. K. S.; Shim K. J.; Wlodkowski S.; 2023; ChatGPT, Can You Generate Solutions for my Coding Exercises? An Evaluation on its Effectiveness in an undergraduate Java Programming Course; In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 54–60. <https://doi.org/10.1145/3587102.3588794>
- [10.] Adamson, V.; Bägerfeldt, J.; 2023; Assessing the effectiveness of ChatGPT in generating Python code (Dissertation). Retrieved from <https://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-22860>

AUTORI · AUTHORS

• **Goran Đambić** – Goran Đambić zaposlen je na Visokom učilištu Algebra kao nastavnik i voditelj Katedre za programsko inženjerstvo. Završio je diplomski studij i magisterij na FER-a, a doktorski studij na Odsjeku za informacijske i komunikacijske znanosti Filozofskog fakulteta u Zagrebu. Područja interesa su mu programsko inženjerstvo s naglaskom na razvoj aplikacija te korištenje metode igrifikacije u provedbi nastavnoga procesa. Predaje na kolegijima Uvod u programiranje, Strukture podataka i algoritmi te Oblikovanje i izrada cjelovitih aplikativnih rješenja.

Korespondencija · Correspondence

goran.dambic@algebra.hr