Решени задачи C:

Предавање 1

1. Пронајди ги и корегирај ги грешките во следните наредби (може да има повеќе од една грешка во секоја од наредбите)

a) scanf(“d”, vrednost); - scanf(“%d”, &vrednost);

b) printf(“Proizvodot na %d i %d e %d”\n, x, y); - printf(“Proizvodot na %d i %d e %d\n”,x,y,x\*y);

c) prvBroj + vtorBroj = sumaBroevi – sumaBroevi = prvBroj + vtorBroj;

d) \*/ Programata naogja maksimum od dva broja/\* - /\* Programata naogja maksimum … \*/

e) Scanf(”%d”, &nekojBroj); - scanf(“%d”, &nekojBroj);

2. Напиши програм кој внесува 4 цели броја и ја испечатува нивната сума.

int number, sum = 0;

for (int i = 0; i < 4; i++) {

scanf("%d", &number);

sum += number;

}

printf("Сумата на броевите е: %d\n", sum);

3. Напиши програм кој ги печати броевите од 1 до 4 на иста линија, при тоа можеш да користиш:

а) една **prinf** наредба, без спецификатори за конверзија - printf("1 2 3 4\n");

б) една **prinf** наредба, со 4 спецификатори за конверзија – printf("%d %d %d %d\n", 1, 2, 3, 4);

в) 4 **printf** наредби. –

printf("1 ");

printf("2 ");

printf("3 ");

printf("4\n");

4. Напиши програм кој вчитува три цели броја и пресметува нивна целобројна средна вредност, сума и производ.

int number, sum = 0, proizvod = 1;

for (int i = 0; i < 3; i++) {

scanf("%d", &number);

sum += number;

proizvod \*= number;

}

printf("Сумата на броевите е: %d\n", sum);

printf("Производ на броевите е: %d\n", prozivod);

printf("Средна вредност на броевите е: %d\n",sum/3);

5. Напиши програм кој внесува петцифрен број, ги распределува сите цифри на бројот во посебни променливи и ги печати секоја од цифрите со растојание една од дуга од три празни места.

Прмер. Нека корисникот внел 42139. Програмот треба да испечати: 4 2 1 3 9

int number, divisor = 100000;

printf("Vnesi 6 cifren broj.\n");

scanf("%d",&number);

while(number > 0) {

printf("%d\t",(number / divisor));

number %= divisor;

divisor /= 10;

}

Предавање 3

**1.** Напиши програм кој како влез зема два цели броја и ги споредува. Потоа го испечатува поголемиот број проследен со коментарот **e pogolem od** и го испечатува и помалиот број. Ако броевите се еднакви, програмот треба да испечати **Broevite se ednakvi**.

int broj1, broj2;

scanf("%d", &broj1);

scanf("%d", &broj2);

if (broj1 > broj2) {

printf("%d e pogolem od %d\n", broj1, broj2);

}else if (broj1 < broj2) {

printf("%d e pogolem od %d\n", broj2, broj1);

} else {

printf("Broevite se ednakvi\n"); }

**2.** Напиши програм кој вчитува два цели броја и одредува дали се парни или непарни.

int broj1, broj2;

scanf("%d", &broj1);

scanf("%d", &broj2);

printf("Првиот број %d е %s\n", broj1, (broj1 % 2 == 0) ? "парен" : "непарен");

printf("Вториот број %d е %s\n", broj2, (broj2 % 2 == 0) ? "парен" : "непарен");

**3.** Обидете се со спецификаторот за конверзија **%d** да испечатите букви на следниот начин:

printf(“%d”, ‘A’) /\* karakteri se koristat vo edinecni navodnici \*/

Што се случува кога се печатат следните карактери: b  c  0  1  2  $  \*  +  / и празно место?

Ќе се испечати ASCII вредноста на А која изнесува 65

**5.** Напиши програма која внесува радиус на круг и испечатува негова плоштина и периметар. За вредност на пи да се земе 3.14.

float radius;

scanf("%f", &radius);

printf("Плоштина на кругот: %.2f\n", 3.14 \* radius \* radius);

printf("Периметар на кругот: %.2f\n", 2 \* 3.14 \* radius);

**6.** Напишете програма која од стандардниот влез зема 5 букви и ги испечатува по следниот редослед: прво петтата, па третата, па втората, па првата, па четвртата буква.

char letters[5];

for (int i = 0; i < 5; i++) {

scanf(" %c", &letters[i]);

}

printf("Se dobi: %c%c%c%c%c\n",

letters[4],

letters[2],

letters[1],

letters[0],

letters[3]);

**8.** Напиши 4 начини за да ја зголемиш вредноста на променливата **x** за 1.

x += 1, x = x + 1, x++ и ++x

Предавање 4

1. Напиши програма која ја пресметува сумата на броевите од 1 до десет.

int suma = 0, broj = 1;

while (broj <= 10) {

suma += broj;

broj++;

}

printf("Сумата на броевите од 1 до 10 е: %d\n", suma);

2. Напиши програма која за внесени десет броја, го пресметува нивниот производ.

int count = 0, broj, proizvod = 1;

while (count < 10) {

scanf("%d", &broj);

proizvod \*= broj;

count++;

}

printf("Производот на внесените броеви е: %d\n", proizvod);

3. Напиши програма која пресметува сума на внесени броеви од тастатура се додека не се внесе -1.

int count = 0, broj = 0, sum = 0;

while (broj != -1) {

sum += broj;

scanf("%d", &broj);

}

printf("Сумата на внесените броеви е: %d\n", sum);

4. Една голема хемиска компанија ги плаќа своите продавачи делумно и на процент. Продавачите земаат плата од 200 евра плус 9% од вкупната нивна продажба за тој месец. На пример, ако некој продавач продал роба вредна 5000 евра, тогаш добива 200 евра плус 9% од 5000, значи вкупно 650 евра. Напиши програма која за внесен месечен промет му ја пресметува платата на вработениот.

float plata = 200.0;

float promet;

while(1) {

printf("Vnesi promet vo evra (-1 za kraj): ");

scanf("%f",&promet);

if(promet == -1)

break;

printf("Platata e: %.2f\n",plata + promet\*0.09);

}

Предавање 5

1.      Да се напише програма која ќе пресметува Питагорини броеви до 100. Тоа се броеви кои го исполнуваат следниот услов a2+b2=c2.

int c\_2, c;

for(int a=0;a<=100;a++) {

for(int b=0;b<=100;b++) {

c\_2 = a\*a + b\*b;

c = (int) sqrt(c\_2);

if(c <= 100 && c \* c == c\_2) {

printf("%d^2 + %d^2 = %d^2\n", a, b, c);

}

}

}

2.      Напиши програм во кој контролата на циклусот на броево од 1 до 100 го прави **for** наредба. Во рамките на наредбата **for** треба да се испечати коментар дека бројот не е делив со три. Доколку бројот е делив со три, не е печати никаков коментар. Задачата да се реши со користење на наредбата **continue**.

for(int i=1;i<=100;i++) {

if(i%3==0)

continue;

printf("%d ne e deliv so 3\n",i);

}

Предавање 6

1. Напиши програм кој конвертира целобројни фарантхајтови температури од 0-212 степени во децимални цезлиусови температури со 3 цифри прецизност. Да се користи формулата:

celsius = 5.0 /9.0 \* (fahrenheit - 32).

Излезот да се испечати во две десно порамнети колони од по 10 карактери, а целзиусовите вредности треба да се предводени од знак и за позитивни и за негативни вредности.

float celsius;

for(int faran=0;faran<=212;faran++) {

celsius = 5.0 / 9.0 \* (faran - 32);

printf("%10d %+.3f\n",faran, celsius);

}

1. Напиши програма која вчитува датуми во облик gg/mm/dd, gg-mm-dd и gg+mm+dd и ги испечаува така да наместо бројот на месецот се испечати имато на месецот.

char \*months[] = {"januari", "februari", "mart", "april", "maj", "juni", "juli", "avgust", "septemvri", "oktomvri", "noemvri", "dekemvri"};

int year,month,day;

scanf("%d%\*[^0-9]%d%\*[^0-9]%d", &year, &month, &day);

printf("%d %s %d godina.\n",day, months[month-1], year);

Предавање 7

1.      Напиши програма која внесува 10 цели броеви и и ги доделува на функцијата **paren** која враќа 1 ако бројот е парен и 0 ако бројот не е парен.

int paren(int i) {

if(i%2==0)

return 1;

return 0;

}

int main() {

for(int i=1;i<11;i++) {

printf("%d\n",paren(i));

}

return 0;

}

2.      Напиши програма која на функцијата **vreme** и доделува три целобројни аргументи, час, минута и секунда и како резултат го враќа бројот на секунди до 12 часот. Имено, ако е 11 часот 59 минути и 58 секунди, тогаш има уште 2 секунди до 12 часот.

int vreme(int hour, int minute, int second) {

if (hour < 12)

return (11 - hour) \* 3600 + (59 - minute) \* 60 + (60 - second);

else if (hour == 12)

return 23\*3600 + (59 - minute) \* 60 + (60 - second);

else

return (23 - (hour-12)) \* 3600 + (59 - minute) \* 60 + (60 - second);

}

int main() {

int hour, minute, second;

scanf("%d%d%d", &hour,&minute,&second);

printf("Ostanato vreme do 12 casot: %d sekundi.\n", vreme(hour, minute, second));

return 0;

}

3.      Напиши програма која на функцијата **obratno** и доделува цел број. Оваа функција го враќа бројот напишан со обратни цифри, т.е. ако сме ја повикале функцијата со параметарот 1243, треба да врати 3421.

int obratno(int broj) {

int obratenBroj = 0;

while (broj > 0) {

int poslednaCifra = broj % 10;

obratenBroj = obratenBroj \* 10 + poslednaCifra;

broj /= 10;

}

return obratenBroj;

}

int main() {

int broj;

scanf("%d",&broj);

printf("%d", obratno(broj));

return 0;

}

4.      Напиши програма во која дефинираш функција **nzd** за пресметување на најголемиот заеднички делител на два броја.

int nzd(int broj1, int broj2) {

while (broj2 != 0) {

int temp = broj2;

broj2 = broj1 % broj2;

broj1 = temp;

}

return broj1;

}

int main() {

int broj1, broj2;

scanf("%d%d", &broj1, &broj2);

printf("NZD na %d i %d е: %d\n", broj1, broj2, nzd(broj1, broj2));

return 0;

}

5.      Напиши итеративна верзија на функцијата за фибоначиеви броеви.

int fib(int n) {

if (n <= 1) {

return n;

}

int fibN\_2 = 0;

int fibN\_1 = 1;

int fibN = 0;

for (int i = 2; i <= n; ++i) {

fibN = fibN\_2 + fibN\_1;

fibN\_2 = fibN\_1;

fibN\_1 = fibN;

}

return fibN;

}

int main() {

int n;

scanf("%d", &n);

printf("%d",fib(n));

return 0;

}

6.      Напиши рекурзивна верзија на функцијата за пресметување најголем заеднички делител.

int nzd(int broj1, int broj2) {

if (broj2 == 0) {

return broj1;

} else {

return nzd(broj2, broj1 % broj2);

}

}

int main() {

int broj1, broj2;

scanf("%d%d", &broj1, &broj2);

printf("NZD na %d i %d е: %d\n", broj1, broj2, nzd(broj1, broj2));

return 0;

}

7.      Напиши рекурзивна верзија на програма која за внесено **n** повикува функција **dzvezdi** со параметар **n** која во зависност од **n** испечатува ѕвезди на следниот начин:

void dzvezdi(int n) {

for(int i=0;i<n;i++) {

printf("\* ");

}

printf("\n");

if(n-1 > 0)

dzvezdi(n-1);

}

int main() {

int n;

scanf("%d",&n);

dzvezdi(n);

return 0;

}

Предавање 8

1.      Напиши програм кој вчитува вредности за температури за секој ден од месецот и наоѓа средна месечна температура. Да се земе предвид дека различни месеци имаат различен број на денови.

int months[] = {31,28,31,30,31,30,31,31,30,31,30,31};

int month;

scanf("%d",&month);

int temp[months[month-1]];

int sum = 0;

for(int i=0;i<months[month-1];i++) {

scanf("%d",&temp[i]);

sum += temp[i];

}

printf("AVG: %f", (float)sum/months[month-1]);

2.      Прости броеви се броеви кои се деливи само со себе си и со бројот 1. Напиши програм кој внесува **n** елементи во низа. Да се овозможи функционалност да имаме променлив број на елементи во низата. Во почетокот сите елементи да се иницијализираат на 1. Потоа да се напише функција **prosti** која секој елемент од низата чиј индекс е прост број, го поставува на 0. На крај од така добиената низа да се најде начин да се испечатат простите броеви до **n**.

void prosti(int niza[], int n) {

for (int i = 2; i < n; i++) {

if (niza[i] == 1) {

for (int j = i \* i; j < n; j += i) {

niza[j] = 0;

}

}

}

}

void pecatiProsti(int niza[], int n) {

for (int i = 2; i < n; i++) {

if (niza[i] == 1)

printf("%d ", i);

}

}

int main() {

int n;

scanf("%d",&n);

int niza[n];

for (int i = 0; i < n; i++) {

niza[i] = 1;

}

prosti(niza, n);

pecatiProsti(niza, n);

return 0;

}

3.      Палиндроми се низи од букви што се читаат исто и оддесно на лево и одлево на десно. На пример зборот “radar” е палиндром. Да се напише програм во кој внесуваме низа од карактери. Треба да имаме функција **palindrom** која враќа вредност 1 ако стрингот е палиндром или 0 инаку.

int palindrom(char \*string) {

for(int i=0;i< strlen(string)/2;i++) {

if(string[i] != string[strlen(string)-1-i])

return 0;

}

return 1;

}

int main() {

char string[100];

scanf("%s", string);

if(palindrom(string) == 1)

printf("palindrom");

else

printf("ne e palindrom");

return 0;

}

5.      Да се напише програма која симулира селекциско сортирање. Селекциското сортирање го бара најмалиот елемент во низата и го заменува со првиот елемент. Сега, на прво место сигурно ни се наоѓа најмалиот елемент во низата. Потоа, во остатокот од низата, од вториот до **n**-тиот елемент, повторно го бараме новиот најмал елемент и му ја заменуваме вредноста со вториот елемент од низата итн. Ова сортирање има слични процесирачки капацитети како сортирање со методот на меурче.

а) Да се напише итеративна верзија на функцијата **selekcija** која го врши селекциското сортирање.

б) Да се напише рекурзивна верзија на селекциското сортирање.

в) Да се одговори зошто процесирачките капацитети на ова сортирање се слични како оние со методот на меурче.

//A)

void sort(int niza[]) {

int min;

for(int i=0;i<11;i++) {

min = i;

for(int j=i+1;j<12;j++) {

if(niza[min] > niza[j]) {

min = j;

}

}

int temp = niza[i];

niza[i] = niza[min];

niza[min] = temp;

}

}

////B)

void sort(int niza[],int i, int j,int min) {

if(i < 12) {

if(j < 12) {

if(niza[min] > niza[j])

min = j;

sort(niza,i,j+1,min);

} else {

int temp = niza[i];

niza[i] = niza[min];

niza[min] = temp;

sort(niza,i+1,i+2,i+1);

}

}

}

//C

void pecati(int niza[]) {

for(int i=0;i<12;i++) {

printf("%d ", niza[i]);

}

}

int main() {

int niza[] = {6,7,453,234,425,64,1,32,54,190,23,45};

sort(niza,0,1,0);

pecati(niza);

return 0;

}

Предавање 9

1. Да се најде обиколката и плоштината на правилен n-аголник.

int n;

float strana;

scanf("%d%f", &n, &strana);

printf("Obikolka: %.2f\n", n \* strana);

printf("Ploshina: %.2f\n", (n \* strana \* strana) / (4 \* tan(M\_PI / n)));

1. Да се најде спротивен број на природниот број n. (Тоа е природниот број што ги има истите цифри со n, напишани во спротивен редослед.

int sprotiven(int n) {

int sprotivenBroj;

while (n > 0) {

int cifra = n % 10;

sprotivenBroj = sprotivenBroj \* 10 + cifra;

n /= 10;

}

return sprotivenBroj;

}

int main() {

int n;

scanf("%d",&n);

printf("%d", sprotiven(n));

return 0;

}

1. Да се најдат совршените броеви помали од природниот број n.

void sovrsenBroj(int n) {

int suma = 0;

for(int i=0;i<n;i++) {

if(n%i == 0)

suma += i;

}

if(suma == n)

printf("%d\n",n);

}

int main() {

int n;

scanf("%d",&n);

for(int i=1;i<=n;i++) {

sovrsenBroj(i);

}

return 0;

}

1. Да се најдат природните броеви во интервалот [n1,n2], кои се деливи со сво­јот спротивен број.

int sprotiven(int n) {

int reversedNumber = 0;

while (n != 0) {

int digit = n % 10;

reversedNumber = reversedNumber \* 10 + digit;

n /= 10;

}

return reversedNumber;

}

int main() {

int n,m;

scanf("%d%d",&n,&m);

for(int i=n;i<m;i++) {

if(i%sprotiven(i) == 0) {

printf("%d\n",i);

}

}

return 0;

}

5. Секоја од n бактерии се дели на две на секои i минути, додека на секои j минути изумираат x% од живите бактерии во тој момент. Колку бактерии ќе има по k минути?

int calculateBacteria(int n, int i, int j, int x, int k) {

int bacteria = n;

for (int time = 0; time <= k; time++) {

if (time % i == 0 && time != 0)

bacteria \*= 2;

if (time % j == 0 && time != 0)

bacteria -= bacteria \* x / 100;

}

return bacteria; }

1. Да се најдат сите растечки поднизи во низата броеви [ai]n. (Растечка подни­за е онаа во која секој елемент е поголем од претходниот.)

int niza[] = {1,6,4,8,4,87,56,34,65,86,665,752,4123,5464,3,4123,42133,66666,66667,77776};

int length = sizeof(niza) / sizeof(niza[0]);

for(int i=1;i<length;i++) {

if(niza[i] < niza[i-1]) {

printf("\n");

}

printf("%d ",niza[i]);

}

1. *Да се најдат сите пилести поднизи во низата броеви [ai]n. (Пилеста подни­за е онаа во која претходниот и следниот елемент на секој елемент се по­мали (поголеми) од него.)*

int niza[] = {1,6,4,8,4,87,56,34,65,86,665,752,4123,5464,3,4123,42133,66666,66667,77776};

int length = sizeof(niza) / sizeof(niza[0]);

for(int i=1;i<length-1;i++) {

if(niza[i] < niza[i-1] && niza[i] < niza[i+1] || niza[i] > niza[i-1] && niza[i] > niza[i+1]) {

printf("%d %d %d\n",niza[i-1],niza[i],niza[i+1]);

}

}

1. За даден природен број n да се испечатат следниве фигури со користење на рекурзија.

void rekurzija(int n, int m) {

if(m > 0) {

for(int i=m;i<=n;i++) {

printf("%d ",i);

}

printf("\n");

rekurzija(n,m-1);

}

if(m>1) {

for(int i=m;i<=n;i++) {

printf("%d ",i);

}

printf("\n");

}

}

int main() {

int n;

scanf("%d",&n);

rekurzija(n,n);

return 0;

}

Предавање 10 проект

1. Да се направи програма во која ќе се сместат оцени на студенти по предмети (дводимензионална низа во која во редици се студентите, а во колони се нивните оцени по 5 предмети). Програмата треба да го нуди следниот кориснички посредник:

Vnesi izbor:

0 Ispechati ja nizata so oceni

1 Najdi ja maksimalnataocena

2 Najdi ja minimalnata ocena

3 Ispecati ja prosechnata ocena za sekoja student

4 Kraj

Програмата да се реши со користење на покажувачи до функциите **max**, **min**, **pecatiNiza**, **рrosek**. Покажувачите кон четирите функции да се сместат во низата **obrabotiOceni**, а како индекс за повикување на секоја од функкциите да се користи изборот од корисничкото мени.

void pecati(int array[5][5]) {

for(int i=0;i<5;i++) {

for(int j=0;j<5;j++) {

printf("%d ",array[i][j]);

}

printf("\n");

}

}

void maxOcena(int array[5][5]) {

for(int i=0;i<5;i++) {

int max = 0;

for(int j=0;j<5;j++) {

if(array[i][j] > max)

max = array[i][j];

}

printf("%d ",max);

}

}

void minOcena(int array[5][5]) {

for(int i=0;i<5;i++) {

int min = 5;

for(int j=0;j<5;j++) {

if(array[i][j] < min)

min = array[i][j];

}

printf("%d ",min);

}

}

void prosecnaOcena(int array[5][5]) {

for(int i=0;i<5;i++) {

int sum = 0;

for(int j=0;j<5;j++) {

sum += array[i][j];

}

printf("%.2f ",sum/5.0);

}

}

int main()

{

int oceni[5][5] = {{5,3,4,5,2},{4,2,5,4,5},{3,5,5,4,5},{2,4,4,5,3},{5,5,5,5,5}};

void (\*f[4])( int[5][5] ) = { pecati, maxOcena, minOcena, prosecnaOcena };

int choice;

printf( "Vnesi broj megju 0,1,2,3 i 4 za kraj: " );

scanf( "%d", &choice );

while ( choice >= 0 && choice < 4 ) {

(\*f[ choice ])( oceni );

printf( "\nVnesi broj megju 0,1,2,3 i 4 za kraj: " );

scanf( "%d", &choice );

}

printf( "Izvrshuvanjeto na programata zavrshi.\n" );

return 0;

}

Предавање 11

1. Напиши програм кој внесува телефонски број како стринг во облик (333) 333-3333. Програмот треба со користење на наредбата **strtok** да го извлече кодот на државата како едно делче, а потоа да го спои целиот телефонски број во друг стринг без -. Програмот треба да го конвертира бројот на државата во **int**, а седмоцифрениот телефонски број во **long**. Двата бројат треба да бидат испечатени.

char phone\_number[14];

scanf("%s",phone\_number);

char \*token;

char \*delimiters = "()-";

char country\_code[4];

char phone\_digits[11];

token = strtok(phone\_number, delimiters);

strncpy(country\_code, token, sizeof(country\_code) - 1);

country\_code[3] = '\0';

while ((token = strtok(NULL, delimiters)) != NULL) {

strncat(phone\_digits, token, sizeof(phone\_digits) - strlen(phone\_digits) - 1);

}

int country\_code\_int = atoi(country\_code);

long phone\_digits\_long = atol(phone\_digits);

printf("Kod na drzavata: %d\n", country\_code\_int);

printf("Telefonski broj: %ld\n", phone\_digits\_long);

return 0;

2. Да се пронајдат на Интернет кодовите за бројки и букви во Морзеовата азбука. Да се напише програм кој чита текст фраза и ја дава нејзината вредност во Морзеовата азбука. Да се направи и друга функција на која и се предава Морзеов текст, а треба како резултат да се добие текст фраза во говорен јазик.

void textToMorse(char \*text) {

char \*morseCodes[] = {".-", "-...", "-.-.", "-..", ".", "..-.", "--.", "....", "..", ".---",

"-.-", ".-..", "--", "-.", "---", ".--.", "--.-", ".-.", "...", "-",

"..-", "...-", ".--", "-..-", "-.--", "--..",

"-----", ".----", "..---", "...--", "....-", ".....", "-....", "--...", "---..", "----."};

for (int i = 0; text[i]; i++) {

if (isalpha(text[i])) {

char upperChar = toupper(text[i]);

printf("%s ", morseCodes[upperChar - 'A']);

} else if (isdigit(text[i])) {

printf("%s ", morseCodes[text[i] - '0' + 25]);

} else if (text[i] == ' ') {

printf(" ");

}

}

printf("\n");

}

void morseToText(char \*morse) {

char \*morseCodes[] = {".-", "-...", "-.-.", "-..", ".", "..-.", "--.", "....", "..", ".---",

"-.-", ".-..", "--", "-.", "---", ".--.", "--.-", ".-.", "...", "-",

"..-", "...-", ".--", "-..-", "-.--", "--..",

"-----", ".----", "..---", "...--", "....-", ".....", "-....", "--...", "---..", "----."};

char \*token = strtok(morse, " ");

while (token != NULL) {

for (int i = 0; i < 36; i++) {

if (strcmp(token, morseCodes[i]) == 0) {

if (i < 26) {

printf("%c", 'A' + i);

} else {

printf("%c", '0' + (i - 25));

}

break;

}

}

token = strtok(NULL, " ");

}

printf("\n");

}

int main()

{

char input[100];

//Vnesi tekst

fgets(input, sizeof(input), stdin);

if (input[strlen(input) - 1] == '\n') {

input[strlen(input) - 1] = '\0';

}

//Morzeov tekst

textToMorse(input);

//Vnesi morzeov tekst

fgets(input, sizeof(input), stdin);

if (input[strlen(input) - 1] == '\n') {

input[strlen(input) - 1] = '\0';

}

//Obicen tekst

morseToText(input);

return 0;

}

Предавање 12

1. Само со користење на операторот за шифтирање да се напише функција **mnoziSoStepenOdDva** која зема два целобројни аргументи: **broj** и **stepen** и ја пресметува следната вредност:

int broj, stepen, rezultat;

scanf("%d%d",&broj,&stepen);

rezultat = broj << stepen;

printf("%d", rezultat);

2. Напиши програм кој го превртува редоследот на битовите на целобројна **unsigned** вредност. Програмот треба да вчита вредност од корисникот и да ја повика функцијата **prevtriBitovi** за да се испечатат битовите во обратен редослед. Вредностите на битовите да се испечатат и пред и после превртувањето.

void pecatiBajti(unsigned broj) {

for(int i=31;i>=0;i--) {

printf("%d", (broj >> i) & 1);

}

}

void prevrtiBitovi(unsigned broj) {

for(int i=0;i<31;i++) {

printf("%d", (broj >> i) & 1);

}

}

int main() {

unsigned broj;

scanf("%u", &broj);

pecatiBajti(broj);

printf("\n");

prevrtiBitovi(broj);

return 0;

}

Предавање 13

1. Напиши функција **pecatiListaNanazad** која рекурзивно ги печати елементите од листа наназад. Елементите на листата се состојат од едно целоборјно поле **broj** и еден карактер кој се вика **karakter**.

struct lista {

int broj;

char karakter;

struct lista \*nextPtr;

};

typedef struct lista Lista;

typedef Lista \*ListNodePtr;

void insert( ListNodePtr \*sPtr, int broj, char karakter ) {

ListNodePtr newPtr;

ListNodePtr previousPtr;

ListNodePtr currentPtr;

newPtr = malloc(sizeof(Lista));

if(newPtr != NULL) {

newPtr->broj = broj;

newPtr->karakter = karakter;

previousPtr = NULL;

currentPtr = \*sPtr;

while(currentPtr != NULL && broj > currentPtr->broj) {

previousPtr = currentPtr;

currentPtr = currentPtr->nextPtr;

}

if(previousPtr == NULL) {

newPtr->nextPtr = \*sPtr;

\*sPtr = newPtr;

} else {

previousPtr->nextPtr = newPtr;

newPtr->nextPtr = currentPtr;

}

}

}

void pecati(ListNodePtr \*sPtr) {

ListNodePtr currentPtr = \*sPtr;

if (currentPtr == NULL) {

return;

} else {

pecati(&currentPtr->nextPtr);

printf("%d,%c\n", currentPtr->broj, currentPtr->karakter);

}

}

int main()

{

ListNodePtr startPtr = NULL;

int choice = 1,broj;

char karakter;

printf( "? " );

scanf( "%d", &choice );

while ( choice != 3 ) {

printf( "Vnesi karakter: \n" );

scanf("%d",&broj);

scanf(" %c",&karakter);

insert( &startPtr,broj,karakter );

printf( "? " );

scanf( "%d", &choice );

}

pecati(&startPtr);

return 0;

}

1. Напиши програма која внесува елементи со функцијата **vnesi** несортирано, на крајот на листата, ја користи функцијата **sortOpagjanje** која сортира елементи во опаѓачки редослед, функцијата **sortRastenje** која ги сортира елемнтите во растечки редослед, **brisi** која брише елемент со дадена содржина. Елементите на листата се целобројни елементи.

struct lista {

int number;

struct lista \*nextPtr;

};

typedef struct lista Lista;

typedef Lista \*ListNodePtr;

void insert( ListNodePtr \*sPtr, int number ) {

ListNodePtr newPtr;

ListNodePtr previousPtr;

ListNodePtr currentPtr;

newPtr = malloc(sizeof(Lista));

if(newPtr != NULL) {

newPtr->number = number;

previousPtr = NULL;

currentPtr = \*sPtr;

while(currentPtr != NULL) {

previousPtr = currentPtr;

currentPtr = currentPtr->nextPtr;

}

if(previousPtr == NULL) {

newPtr->nextPtr = \*sPtr;

\*sPtr = newPtr;

} else {

previousPtr->nextPtr = newPtr;

newPtr->nextPtr = currentPtr;

}

}

}

void pecati(ListNodePtr \*sPtr) {

ListNodePtr currentPtr = \*sPtr;

if (currentPtr == NULL) {

return;

} else {

pecati(&currentPtr->nextPtr);

printf("%d ---> ", currentPtr->number);

}

}

void sortMaxToMin(ListNodePtr \*sPtr) {

ListNodePtr currentPtr, nextNode, temp;

int swapped;

if (\*sPtr == NULL || (\*sPtr)->nextPtr == NULL) {

return;

}

do {

swapped = 0;

currentPtr = \*sPtr;

temp = NULL;

while (currentPtr->nextPtr != NULL) {

nextNode = currentPtr->nextPtr;

if (currentPtr->number > nextNode->number) {

if (temp == NULL) {

\*sPtr = nextNode;

} else {

temp->nextPtr = nextNode;

}

currentPtr->nextPtr = nextNode->nextPtr;

nextNode->nextPtr = currentPtr;

temp = nextNode;

swapped = 1;

} else {

temp = currentPtr;

currentPtr = currentPtr->nextPtr;

}

}

} while (swapped);

}

void sortMinToMax(ListNodePtr \*sPtr) {

ListNodePtr currentPtr, nextNode, temp;

int swapped;

if (\*sPtr == NULL || (\*sPtr)->nextPtr == NULL) {

return;

}

do {

swapped = 0;

currentPtr = \*sPtr;

temp = NULL;

while (currentPtr->nextPtr != NULL) {

nextNode = currentPtr->nextPtr;

if (currentPtr->number < nextNode->number) {

if (temp == NULL) {

\*sPtr = nextNode;

} else {

temp->nextPtr = nextNode;

}

currentPtr->nextPtr = nextNode->nextPtr;

nextNode->nextPtr = currentPtr;

temp = nextNode;

swapped = 1;

} else {

temp = currentPtr;

currentPtr = currentPtr->nextPtr;

}

}

} while (swapped);

}

int main()

{

ListNodePtr startPtr = NULL;

int choice = 1;

int number;

while ( choice != 3 ) {

printf( "Vnesi karakter: \n" );

scanf("%d",&number);

insert( &startPtr,number );

printf( "? " );

scanf( "%d", &choice );

}

sortMaxToMin(&startPtr);

pecati(&startPtr);

printf("\n");

sortMinToMax(&startPtr);

pecati(&startPtr);

return 0;

}

1. Напиши програм кој внесува линија текст и користи стек за да се испечати текстот во обратен дерослед. Текстот да содржи и празни места.

struct stackNode {

char data;

struct stackNode \*nextPtr;

};

typedef struct stackNode StackNode;

typedef StackNode \*StackNodePtr;

void push( StackNodePtr \*topPtr, char info );

int pop( StackNodePtr \*topPtr );

int isEmpty( StackNodePtr topPtr );

void printStack( StackNodePtr currentPtr );

void instructions( void );

int main()

{

StackNodePtr stackPtr = NULL;

int choice;

char value;

char string[100];

printf("Vnesi string: ");

scanf(" %[^\n]", string);

for (int i = 0; i < strlen(string); i++) {

push(&stackPtr, string[i]);

}

for (int i = 0; i < strlen(string); i++) {

printf("%c",pop(&stackPtr));

}

return 0;

}

void instructions( void )

{

printf( "Vnesi izbor:\n"

"1 za stavanje vrednost na stek\n"

"2 za vadenje vrednost od stek\n"

"3 za kraj na programot\n" );

}

void push( StackNodePtr \*topPtr, char info )

{

StackNodePtr newPtr;

newPtr = malloc( sizeof( StackNode ) );

if ( newPtr != NULL ) {

newPtr->data = info;

newPtr->nextPtr = \*topPtr;

\*topPtr = newPtr;

}

else {

printf( "%c ne e vnesen. Nema dovolno memorija.\n", info );

}

}

int pop( StackNodePtr \*topPtr )

{

StackNodePtr tempPtr;

char popValue;

tempPtr = \*topPtr;

popValue = ( \*topPtr )->data;

\*topPtr = ( \*topPtr )->nextPtr;

free( tempPtr );

return popValue;

}

void printStack( StackNodePtr currentPtr )

{

if ( currentPtr == NULL ) {

printf( "Stekot e prazen.\n\n" );

} /\* kraj if \*/

else {

printf( "Stekot e:\n" );

while ( currentPtr != NULL ) {

printf( "%c --> ", currentPtr->data );

currentPtr = currentPtr->nextPtr;

}

printf( "NULL\n\n" );

}

}

int isEmpty( StackNodePtr topPtr )

{

return topPtr == NULL;

}