



დარჩენილი დრო 0:49:58

დამალე



კითხვა 1

ჯერ პასუხგაცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

დახარისხების რომელი ხერხია გამოყენებული მოცემულ პროგრამაში

```
int main ()
{
    const int arraySize = 10;
    int A[arraySize] = {2, 6, 4, 8, 19, 100, 89, 68, 45, 37};
    int hold;
    cout << მასივის ელემენტები საწყისი თანმიმდევრობით:" << "\n ";
    for (int i = 0; i < arraySize; i++)
        cout << A [i] << " ";

    for(int pass = 1; pass < arraySize; pass++)
        for (int i = 0; i < arraySize-pass; i++)
            if (A[i] > A[i + 1]) {
                hold = A[i];
                A[i] = A[i + 1];
                A[i + 1] = hold;
            }

    cout << " \n დახარისხებული მასივის ელემენტები \n ";
    for (int i = 0; i < arraySize; i++)
        cout << A[i] << " ";
    cout << " \n ";
    getch();
    return 0;
}
```

აირჩიეთ ერთი:

- ☐ a. კომბინირებული ალგორითმი
- ☐ b. არც ერთი
- ☐ c. ბუშტისებრი-ჩაძირვის ალგორითმი
- ☐ d. დახარისხება სელექციის მეთოდით



კითხვა 2

ჯერ პასუხგაცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

რას აკეთებს პროგრამის შემდეგი ფრაგმენტი?

```
for (i=0;i<4;i++)  
{  
  for (pass = 1; pass < 3; pass++)  
    for (j = 0; j <2; j++)  
      if (A[i][j]>A[i][j+1])  
      {  
        hold = A[i][j];  
        A[i][j]=A[i][j+1];  
        A[i][j+1]=hold;  
      }  
}
```

აირჩიეთ ერთი:

- ☐ a. მატრიცის თითოეულ სვეტის ელემენტებს ალაგებს კლებადობით
- ☐ b. მატრიცის თითოეული სტრიქონის ელემენტებს ალაგებს კლებადობის მიხედვით
- ☐ c. მატრიცის თითოეულ სვეტის ელემენტებს ალაგებს ზრდადობით
- ☐ d. მატრიცის თითოეული სტრიქონის ელემენტებს ალაგებს ზრდადობის მიხედვით



კითხვა 3

ჯერ პასუხგაუცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

1024 ელემენტიან დახარისხებულ მასივში ორობითი ძეგნის დროს საძიებელი სიდიდის მოსაძებნად საჭიროა

აირჩიეთ ერთი:

- ☐ a. საძიებელ სიდიდესთან არანაკლებ 1024 შედარება
- ☐ b. საძიებელ სიდიდესთან არაუმეტეს 512 შედარება
- ☐ c. საძიებელ სიდიდესთან არაუმეტეს 20 შედარება
- ☐ d. საძიებელ სიდიდესთან არაუმეტეს 10 შედარება



კითხვა 4

ჯერ პასუხაუცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

მოცემულია პროგრამა:

```
int main()
{
    const int arraySize = 10;
    int A[arraySize] = {2, 4, 6, 8, 19, 37, 45, 68, 89, 100};
    int i, low = 0, high = arraySize, middle, searchKey, iKey,
    count = 0;
    cout << " შემოიტანეთ საძიებელი მნიშვნელობა \n ";
    cin >> searchKey;
    while (low <= high)
    {
        middle = (low + high) / 2;
        if (searchKey == A[middle]) {
            iKey = middle;
            count++;
            break;
        }
        if (searchKey < A[middle])
            high = middle - 1; /* მასივის ზედა ზღვარი */

        else if (searchKey > A[middle])
            low = middle + 1; /* მასივის ქვედა ზღვარი */

    }

    if (count != 0)
        cout << " მასივში მოცემული მნიშვნელობის რიგითი ნომერია - " << iKey;
    else
        cout << " მასივში საძიებელი მნიშვნელობის ტოლი ელემენტი არ არის ! \n ";
    getch();
    return 0;
}
```

ციკლის რომელ ბიჯზე მოხდება საძიებელი მნიშვნელობის რიგითი ნომრის მოძებნა თუ searchKey ცვლადის მნიშვნელობას შეიტანთ 37-ს

პირჩიეთ პასუხი:

- ☐ a. მე-3 ბიჯზე
- ☐ b. მე- 2 ბიჯზე
- ☐ c. პირველ ბიჯზე
- ☐ d. მე- 5 ბიჯზე

კითხვა 5

ჯერ პასუხაუცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

ჩამოთვლილთაგან რომელია strcpy ფუნქციის პარამეტრები?

აირჩიეთ ერთი:

- ☐ a. ორი მთელი რიცხვი;
- ☐ b. სტრიქონი და მთელი რიცხვი;
- ☐ c. ორი სტრიქონი.



კითხვა 6

ჯერ პასუხაუცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

რა მოქმედებას ასრულებენ ფუნქციები strcpy (s1,s2) და strcpy_s (s1,s2) ?

აირჩიეთ ერთი:

- ☐ a. შეადარებს s1 და s2 სტრიქონებს ერთმანეთს
- ☐ b. s1-ში ჩაწერს s2 სტრიქონის სიგრძეს
- ☐ c. s2 სტრიქონის კოპირებას s1-ში
- ☐ d. გადააბამს s1 და s2 სტრიქონებს ერთმანეთს და დააბრუნებს მას ფუნქციის მნიშვნელობად

კითხვა 7

ჯერ პასუხაუცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

რა მოქმედებას ასრულებენ ფუნქციები strcat (s1,s2) და strcat_s (s1,s2) ?

აირჩიეთ ერთი:

- ☐ a. s1 სტრიქონიდან ამოჭრის სიმბოლოთა s2 მიმდევრობას
- ☐ b. s2 სტრიქონის კოპირებას s1-ში
- ☐ c. s1 სტრიქონიდან ამოჭრის s2-ში შემავალ სიმბოლოებს
- ☐ d. s1 სტრიქონს მიუერთებს s2-ს



კითხვა 8

ჯერ პასუხგაცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

რა მოქმედებას ასრულებენ ფუნქციები strcmp (s1,s2) და strcmp_s (s1,s2) ?

აირჩიეთ ერთი:

- ☐ a. s1 სტრიქონიდან ამოჭრის სიმბოლოთა s2 მიმდევრობას
- ☐ b. ადარებს ერთმანეთს s1 და s2 სტრიქონებს
- ☐ c. s2 სტრიქონის კოპირებას s1-ში
- ☐ d. s1 სტრიქონზე ბოლოში მიაბამს s2-ს

კითხვა 9

ჯერ პასუხგაცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

რომელი თავის ფაილის ჩართვაა საჭირო პროგრამაში სტრიქონების დამუშავების ბიბლიოთეკური ფუნქციების (strcpy (), strcat ())და სხვა) გამოსაყენებლად?

აირჩიეთ ერთი:

- ☐ a. stdlib
- ☐ b. iostream
- ☐ c. iomanip
- ☐ d. cstring



კითხვა 10

ჯერ პასუხგაცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

რა დაიბეჭდება მოცემული ფრაგმენტის შესრულების შედეგად:

`char S1[10]="ABC";``char S2[10]="abc";``cout<<strcmp(S1,S2);`**აირჩიეთ ერთი:**☐ a. -1☐ b. 1☐ c. 0**კითხვა 11**

ჯერ პასუხგაცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

რა დაიბეჭდება მოცემული ფრაგმენტის შესრულების შედეგად:

`char b[20]="information";``cout<<strlen(b);`

Answer:



კითხვა 12**ჯერ პასუხგაუცემელი****გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00**

v1 (latest)

რას შეასრულებს შემდეგი ფრაგმენტი?

```
const int N=10;  
int a[N], i;  
for (i=0; i<N; i++)  
    a[i]=rand()%31;
```

აირჩიეთ ერთი:

- ☐ a. 0-დან 31-ჩათვლით ყველა რიცხვს შეიტანს მასივში;
- ☐ b. 0-დან 30-მდე ყველა რიცხვს შეიტანს მასივში;
- ☐ c. [0, 30] დიაპაზონიდან ალებულ შემთხვევით რიცხვებს შეიტანს მასივში;
- ☐ d. [0,31]დიაპაზონიდან ალებულ შემთხვევით რიცხვებს შეიტანს მასივში;



კითხვა 13

ჯერ პასუხგაუცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

რა დაიბეჭდება ეკრანზე შემდეგი ფრაგმენტის შესრულების შემდეგ?

```
int main()
{
    deque <int> dq;
    cout << dq.empty() << endl;
    dq.push_front(3);
    dq.push_front(2);
    dq.push_back(4);
    dq.push_front(1);
    dq.push_back(5);
    cout << boolalpha << dq.empty() << endl;
    cout << dq.size() << endl;
    cout << dq.front() << endl;
    cout << dq.back() << endl;
}
```

პირველი პასუხი:☐ a.

1
false
5
1
5

☐ b.

0
false
5
1
5

☐ c.

true
false
5



1

5

☐ d.

false

true

5

1

5

კითხვა 14

ჯერ პასუხაუცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

მოცემულია პროგრამის ფრაგმენტი:

```
{  
dq.push_front(22);  
dq.push_back(55);  
dq.push_back(66);  
}
```

რამდენი ელემენტია მოცემულ დეკში?

Answer:



კითხვა 15

ჯერ პასუხაუცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

მოცემულია პროგრამის ფრაგმენტი:

```
{  
dq.push_front(23);  
dq.push_back(55);  
dq.push_back(66);  
cout<<dq.front()<<endl;  
}
```

რა დაიბეჭდება მოცემული ფრაგმენტის შესრულების შემდეგ?

Answer:



კითხვა 16

ჯერ პასუხგაუცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

რა დაიბეჭდება ეკრანზე შემდეგი ფრაგმენტის შესრულების შემდეგ?

```
int main()
{
    stack <int> st;
    cout << boolalpha << st.empty() << endl;
    st.push(5);
    st.push(7);
    st.push(9);
    cout<<st.size()<<endl;
    cout<<st.top()<<endl;
    st.pop();
    cout<<st.size()<<endl;
    st.pop();
    cout<<st.size()<<endl;
    cout << boolalpha << st.empty() << endl;
    st.push(11);
    st.push(13);
    cout<<st.size()<<endl;
    return 0;
}
```

პირველი პრთი:☐ a.

true

3

5

2

1

false

3

☐ b.

false

3

9



2

1

true

3

☐ c.

false

3

5

2

1

true

3

☐ d.

true

3

9

2

1

false

3



კითხვა 17

ჯერ პასუხაუცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

მოცემულია პროგრამის ფრაგმენტი:

{

.....

st.push(22);

st.push(2);

st.push (19);

.....

}

რამდენი ელემენტი მოცემულ სტეკში?

Answer:

კითხვა 18

ჯერ პასუხაუცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

მოცემულია პროგრამის ფრაგმენტი:

{

.....

st.push(14);

st.push(2);

st.push (19);

.....

}

რომელია პირველი ელემენტი სტეკში?



Answer:

კითხვა 19

ჯერ პასუხაუცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

რა დაიბეჭდება ეკრანზე შემდეგი ფრაგმენტის შესრულების შემდეგ?

```
{  
queue <int> q;  
cout << boolalpha << q.empty() << endl;  
q.push(17);  
q.push(25);  
q.push(9);  
cout<<q.size()<<endl;  
q.pop();  
cout << boolalpha << q.empty() << endl;  
cout<<q.front()<<endl;  
cout<<q.back()<<endl;  
q.push(11);  
cout<<q.size()<<endl;  
}
```

პირველი პრტი:☐ a.

false

3

true

25

9

3

☐ b.

true

3

false

25

9

3

☐ c.

true

3

false

9

25

3

☐ d.

false

3

true

9

25

3

კითხვა 20

ჯერ პასუხაუცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

მოცემულია პროგრამის ფრაგმენტი:

```
{  
queue <int> q;  
q.push(95);  
q.push(22);  
q.push(27);  
q.push(45);  
q.push(52);  
q.pop();  
cout<<q.size()<<endl;  
}
```

რა დაიბეჭდება მოცემული ფრაგმენტის შესრულების შემდეგ?

Answer:



კითხვა 21

ჯერ პასუხგაცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

მოცემულია პროგრამის ფრაგმენტი:

```
{  
queue <int> q;  
q.push(95);  
q.push(22);  
q.push(27);  
q.push(45);  
q.push(52);  
q.pop();  
cout<<q.front()<<endl;  
}
```

რა დაიბეჭდება მოცემული ფრაგმენტის შესრულების შემდეგ?

Answer:

კითხვა 22

ჯერ პასუხგაცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

რა დაიბეჭდება მოცემული ფრაგმენტის შესრულების შედეგად?

```
int n =179;  
int *p;  
p = &n;  
cout<<*p;
```



Answer:

კითხვა 23

ჯერ პასუხგაუცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

გვაქვს `int b[3] = { 2, 4, 6 }, *bPtr=b;` რომელი ოპერატორი მოგვცემს შეცდომას?

აირჩიეთ ერთი:

- ☐ a. `bPtr = b+1;`
- ☐ b. `bPtr++;`
- ☐ c. `bPtr = &b[0];`
- ☐ d. `b++;`



კითხვა 24

ჯერ პასუხგაუცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

რა შედეგს გამოიტანს შემდეგი პროგრამა:

```
void swap (int *, int *);  
int main(){  
    int a=15, b=25;  
    swap(&a, &b);  
    cout<<a<<b;  
  
}  
void swap (int *x, int *y){  
    int dam;  
    dam = *x;  
    *x = *y;  
    *y = dam;  
}
```

პირველი პასუხი:

- ☐ a. 15 25
- ☐ b. -15 -25
- ☐ c. 25 15
- ☐ d. პროგრამა არ იმუშავებს;



კითხვა 25

ჯერ პასუხგაუცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

მოცემულია ცვლადების მისამართები: &a=63384,&b=64390,&c=64400. რა შედეგს გამოიტანს პროგრამა 32-ბიტის კომპიუტერებისთვის (int - იკავებს 4 ბაიტს, float იკავებს - 4 ბაიტს, double - 8 ბაიტს) :

```
int main()
{
float a,*p1;
int b,*p2;
double c,*p3;
a=2.5; b=3; c=2.96;
p1=&a; p2=&b; p3=&c;
p1++; p2++; p3++;
cout<<"p1="<<p1<<" p2="<<p2<<" p3="<<p3;
}
```

აირჩიეთ ერთი:

- ☐ a. p1=63388 p2=64394 p3=64404
- ☐ b. არცერთი პასუხი არ არის სწორი
- ☐ c. p1=3.5 p2=4 p3=B
- ☐ d. p1=63388 p2=64394 p3=64408



კითხვა 26

ჯერ პასუხგაცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

რა დაიბეჭდება პროგრამის შესრულების შემდეგ:

```
#include <iostream>
using namespace std;
int main()
{
    int *p, x=5;
    p=&x;
    cout<<++(*p);
    system("pause");
    return 0;
}
```

აირჩიეთ ერთი:

- ☐ a. x ცვლადის მისამართი
- ☐ b. 7
- ☐ c. 6
- ☐ d. 5
- ☐ e. კომპილაციის შეცდომაა



კითხვა 27

ჯერ პასუხგაუცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

მიმთითებლის რომელი აღწერაა სწორია?

აირჩიეთ ერთი:

- ☐ a. `int *x;`
- ☐ b. `int x;`
- ☐ c. `ptr x;`
- ☐ d. `int &x;`



კითხვა 28

ჯერ პასუხგაცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

რას ითვლის შემდეგი პროგრამა?:

```
#include <iostream>
using namespace std;
unsigned int f(unsigned int n);
int main()
{
    unsigned int n = 5;
    unsigned int y = f(n);
    cout << n<<"! =" << y << endl;
    return 0;
}
unsigned int f(unsigned int n)
{
    unsigned int t;
    if (n <= 1)
        t = 1;
    else
        t = n * f(n - 1);
    return t;
}
```

პირველი პასუხი:

- ☐ a. ბეჭდავს n-ის შებრუნებულ მნიშვნელობას;
- ☐ b. n-ის ფაქტორიალს;
- ☐ c. n-ის ციფრთა ჯამს;



კითხვა 29

ჯერ პასუხგაუცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

მოცემულია სტრუქტურა, როგორ გამოვაცხადებთ ამ სტრუქტურის ტიპის x ცვლადს?

```
struct rectangle{  
    float length;  
    float width;  
};
```

აირჩიეთ ერთი:

- ☐ a. struct x;
- ☐ b. struct rectangle;
- ☐ c. struct rectangle x;
- ☐ d. struct x rectangle;

კითხვა 30

ჯერ პასუხგაუცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

სტრუქტურა არის მონაცემთა ტიპი, რომელშიც

აირჩიეთ ერთი:

- ☐ a. ელემენტები შეიძლება იყოს სხვადასხვა ტიპის;
- ☐ b. ელემენტები უნდა იყოს მხოლოდ რიცხვითი ტიპის;
- ☐ c. ყველა ელემენტი უნდა იყოს ერთნაირი ტიპის;
- ☐ d. ყველა ელემენტი უნდა იყოს მხოლოდ ტექსტური ტიპის;



კითხვა 31**ჯერ პასუხგაუცემელი****გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00**

v1 (latest)

ფუნქცია ითვლის ნამდვილი რიცხვითი ტიპის მასივის იმ ელემენტების ჯამს, რომელთა ინდექსები იცვლება p-დან q-მდე (იგულისხმება, რომ $p < m$ და $q < m$). რომელი ბრძანებით შეიძლება მოცემული ფუნქციის გამოძახება?

```
const int m =5;
float sum(float [],int , int );
int main()
{
    float mass[m]; int i,p,q;
    float rez;
    for (i=0;i<m;i++) cin>>mass[i];
    cout<<"p?q?";
    cin>>p>>q;
    .....
}
float sum(float a[],int p1,int q1){
    int k; float s =0;
    for(k=p1; k<=q1; k++)
        s +=a[k];
    return s;
}
```

აირჩიეთ ერთი:

- ☐ a. rez = sum(a[],p,q);
- ☐ b. rez = sum(mass[],p,q);
- ☐ c. rez = sum(a,p,q);
- ☐ d. rez = sum(mass,p,q);



კითხვა 32

ჯერ პასუხგაცემელი

გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00

v1 (latest)

რას აბრუნებს შემდეგი ფუნქცია:

```
double S(double a[], int N)
```

```
{ double sum=0;
```

```
for (int i=1; i<N; i+=2)
```

```
sum+=a[i];
```

```
return sum;
```

```
}
```

აირჩიეთ ერთი:

- ☐ a. ყველა რიცხვის ჯამს
- ☐ b. კენტი ინდექსებიანი წევრების ჯამს;
- ☐ c. ლუწი ინდექსებიანი დადებით წევრების ჯამს;
- ☐ d. კენტი ინდექსებიანი დადებით წევრების ჯამს;
- ☐ e. ლუწი ინდექსებიანი წევრების ჯამს;



კითხვა 33**ჯერ პასუხგაუცემელი****გამოანგარიშებულია შემდეგი რაოდენობიდან 1.00**

v1 (latest)

რა მიიღება შემდეგი ფრაგმენტის შესრულების შედეგად?

```
const int N=5;
void printArray(int a[N]);
int main(){
int mass[N]={10,-1,5,7,2};
printArray(mass);
...
}
void printArray(int a[]){
inti;
for(i=0; i<N; i++)
cout<< a[i]<<" ";
}
```

პირჩიეთ ერთი:

- ☐ a. 10 5 2
- ☐ b. 2 7 5 -1 10
- ☐ c. 10 -1 5 7 2

